

AÑO I - N.º 3

SOFT MAGAZINE

495 ptas.

soft
EASY SCRIPT

Hard
1541 FLASH

Bricodore
Fuente de Alimentación para Commodore II parte

cursos
Basic
código máquina

concursos

M. NOVOA

Juegos del mes

MAZIACS

CRISTALS OF ZONG

MOSAICO

Monser



DIRECTOR
José Nieto Rubio

COORDINADOR
Félix Santamaría

SUPERVISOR SOFTWARE
José Luis Roig

DISEÑO
TRAMA 3

REDACCION
Victoria Aguilar
José Luis Roig

COLABORADORES
José Luis Penalva
Marisol Galindo
Mercedes Galindo
Ramón Jiménez

PORTADA
Mauro Novoa

EDITA
MONSER, S.A.

DIRECTOR EDITORIAL
J. L. Cano Regidor

REDACCION, ADMINISTRACION Y PUBLICIDAD
Argos, 9
28037 MADRID
Tel. 742 72 12 / 96

PUBLICIDAD Y SUSCRIPCIONES
Yolanda Bardillo

FOTOCOMPOSICION
H. Corral, S.L.
Burdeos, 2 - Móstoles

FOTOMECANICA
IMAGEN
Nicolás Morales, 34-3º

IMPRIME
GRAFICAS MARTE, S. A.

DEP. LEGAL
M-29620-1.985

DISTRIBUCION
COEDIS

Se solicitará Control O.J.D.

editorial

¡Hola lectores! espero que los juegos mensuales incluídos en los números anteriores os hayan gustado. Como podréis comprobar esta vez hay tres en lugar de dos; esto se debe a que intentamos daros la mayor cantidad de juegos conservando, en la medida que sea posible el precio de la revista. Los juegos de este mes son: el conocido MAZIACS, CRISTALS OF ZONG y MOSAICO. Espero que no perdais la paciencia con ninguno.

Como siempre, vamos a hacer un breve repaso de nuestras secciones.

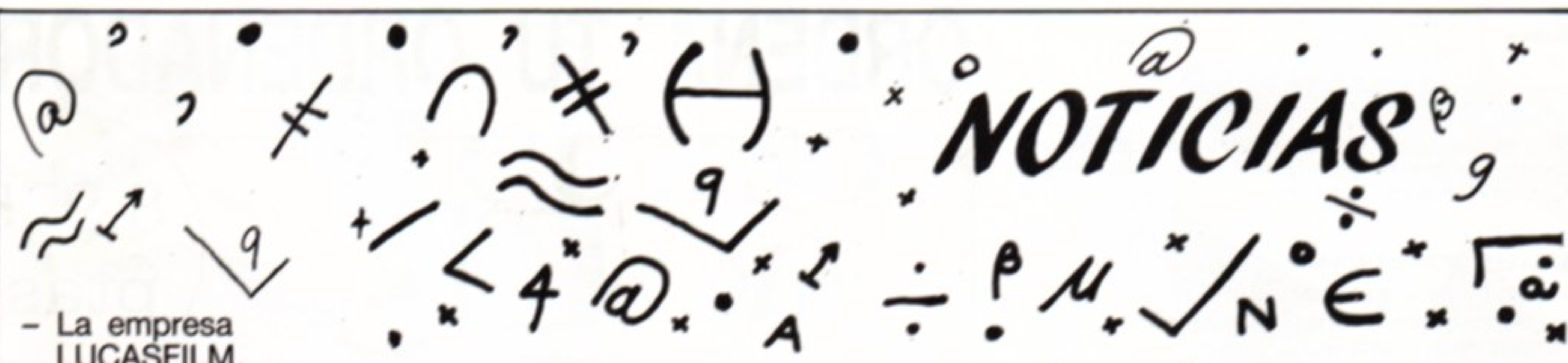
Una de las más interesantes de este número es la de HARD en la que hablaremos del acelerador de discos «¡1541 FLASH!», ya que gracias a él la lenta unidad de disco del Commodore-64 realiza la carga de programas de tres a cinco veces más rápidamente.

En SOFT tendréis un comentario sobre SUPER SCRIPT, un procesador de textos muy útil para escribir vuestras cartas.

En nuestra sección de bricolaje como sabéis «BRICODORE» aparecerá la segunda parte de como construir una fuente de alimentación ininterrumpida para el Commodore.

Por último en este número ya tenemos el ganador de nuestro concurso de preguntas; el resultado del sorteo, realizado entre los acertantes, lo encontraréis en la sección de CONCURSOS.

SUMARIO Nº 3



– La empresa LUCASFILM, creadora de tantas películas como la GUERRA DE LAS GALAXIAS, LOS EWONKS, etc..., ha lanzado al mercado su primer video-juego para el Commodore-64 de la mano de EPIX. Este primer juego de la serie es el «Ballbazer», en el cual tú y tu oponente deberéis luchar por la posesión de un balón en un campo tridimensional. El juego es muy rápido, y los gráficos futuristas. Esperamos que llegue pronto a nuestro país.

— Acaba de aparecer en el mercado mundial el producto «APSOFT-64», que es el BASIC de los ordenadores APPELL II adaptado al C-64 y al C-128. Este paquete contiene:

— Cargador rápido (de 3 a 5 veces).
— Emulador de terminal.
— Base de datos.

Con el APSOFT-64 los usuarios del Commodore tendrán acceso a la extensa librería de programas Basic del APPEL

— La casa de software STARTPOINT ha creado un nuevo «Dos» para la lenta unidad de disco de Commodore-64, la 1541, la cual podrá cargar 10 veces más rápido los programas y salvarlos también 3 veces más rápido. Además contiene:

1. El WEDGE de disco.
2. Un editor de disco.
3. Un miniprocesador de textos.
4. Un duplicador de discos.

y «según dice el fabricante» es 100 % compatible con todo el software existente. Esperamos poder realizar pronto pruebas con este nuevo «DOS» cuyo nombre en el mercado será «STARDOS».



— MICRO ELECTRONICA Y CONTROL ha publicado un catálogo de software existente en España para Commodore. Este catálogo consta de 32 páginas con más de 400 títulos.

— El libro titulado EL CUARTO PROTOCOLO de Frederick Forsyth's va a ser adaptado para el Commodore-64 por la casa HUTCHINSN COMPUTE PUBLISHING con el mismo nombre. El juego consiste en descubrir el lugar donde se oculta una bomba atómica colocada por agentes de la KGB en Londres.



— La impresora DP-100 con interface Comodore se comercializa en España de la mano de EXPOCOM. Esta impresora es una de las recomendables para el Commodore-64.

– La casa FERRE-MORET S.A. ha presentado en SONIMAG nuevos libros del Commodore traducidos al castellano y pertenecientes a la serie DATA BECKER. Son los que a continuación se detallan:

1. Manual escolar para Commodore.
2. El libro de la Ciencia y de la Técnica.
3. Todo sobre el Floppy 1541.
4. Mantenimiento y reparación de 1541.
5. Lenguaje Máquina para el Commodore-64.
6. Diccionario para su Commodore-64.
7. 64 interno.

Otros libros ya presentados de esta misma serie son:

1. Libro de ideas para el Commodore-64.
2. 64 consejos y trucos.
3. Peek's and Pokes.
4. Manual del cassette.



MONSER, S.A.
Calle Argos, 9
28037 MADRID
Telf.: 742 72 12/96

ORDENA TU ORDENADOR

Ahora Vd. puede tener todo su equipo de ordenador en un gabinete de estilo con tres elegantes niveles. No más desórdenes de cables ni de periféricos. Además su equipo estará más protegido.

NO PIERDA ESTA OPORTUNIDAD UNICA

Tendrá espacio a su alcance para hardware y software.

Dispondrá de una unidad de puente de 56,5 cm ancho, 17 cm de alto y 30,5 cm de fondo para su televisor o monitor.

Debajo de esta unidad hay espacio suficiente para guardar su ordenador, aparato de cassette o microdrive.

En una tercera unidad tiene amplio espacio para guardar cintas, diskettes, joysticks, revistas, libros, etc.

Se vende desarmado en una caja plana, es muy fácil de armar, utilizando solamente una llave ALLEN.

El gabinete se presenta en dos colores, NOGAL y ROBLE y tiene dimensiones que se ajustan a las necesidades de espacio y altura que Vd. requiere.

ANCHO 85,5 cm. • ALTO 79,5 cm. • FONDO 60 cm.

Y ADEMAS LOS INTERFACES PARA SU JOYSTICK, IMPRESORA O MICRODRIVE

Interface DK'Tronics
Doble salida en la parte superior. La primera para joystick tipo Kempston y la segunda para software con teclas 6, 7, 8, 9 y 0 o redefinición de teclas.
Ref. 30001. P.V.P. 3.760 ptas.

Interface Centronic.
Para impresora y microdrive en paralelo. Ref. 30010.
P.V.P. 11.358 ptas.

Para pedidos simplemente rellene el cupón.



LOS JOYSTICKS DE GRAN RESPUESTA Y DURABILIDAD









Por favor, envíenme:(marco con una X):

	REF.	COLOR	CANTIDAD	PRECIO	SUBTOTAL
GABINETE	40005	Nogal		**8.975**	
	40005	Roble		**8.975**	
JOYSTICK	30007			**1.900**	
	30003			**2.598**	
	30008			**3.724**	
	30005			**2.450**	
INTERFACE	30001 30010			**3.760** **11.358**	

Gastos de envío gabinete 800 ptas., joysticks 200 ptas. ....

TOTAL ....

Forma de pago:

☐ Talón bancario a nombre de MONSER, S.A.

☐ Giro postal núm. ....

☐ Contra reembolso

Nombre ....

Domicilio ....

Ciudad.................................... Provincia ....

C.P. .... Telf.: ....

FIRMA



Los arrays funcionan de la misma forma que los strings. En este caso tendríamos un array llamado VENTAS, con 6 elementos, es decir, uno por cada mes. Para referirnos a las ventas del mes de Marzo seguiríamos la misma lógica que en el caso de los archivadores (ver fig. 1). Primero haríamos referencia al array de ventas (VENTAS), a continuación nos referiríamos al número que ocupa dentro del array. A este número se le lláma índice y debe ser un número o variable entera, o bien una expresión que de como resultado un número entero. Así tendremos:

VENTAS (0) = Ventas hechas en Enero.
VENTAS (1) = Ventas hechas en Febrero.
VENTAS (2) = Ventas hechas en Marzo.
VENTAS (3) = Ventas hechas en Abríl.
VENTAS (4) = Ventas hechas en Mayo.
VENTAS (5) = Ventas hechas en Junio.

Observamos que el índice comienza en 0. Gráficamente quedará así:

VENTAS (0)	
VENTAS (1)	
VENTAS (2)	
VENTAS (3)	
VENTAS (4)	
VENTAS (5)	

Los nombres de los array siguen las mismas normas que las de las variables, ya que al fin y al cabo una variable se puede considerar como un array unidimensional con un solo elemento. Por ejemplo:
A % y A % (0) representan lo mismo.

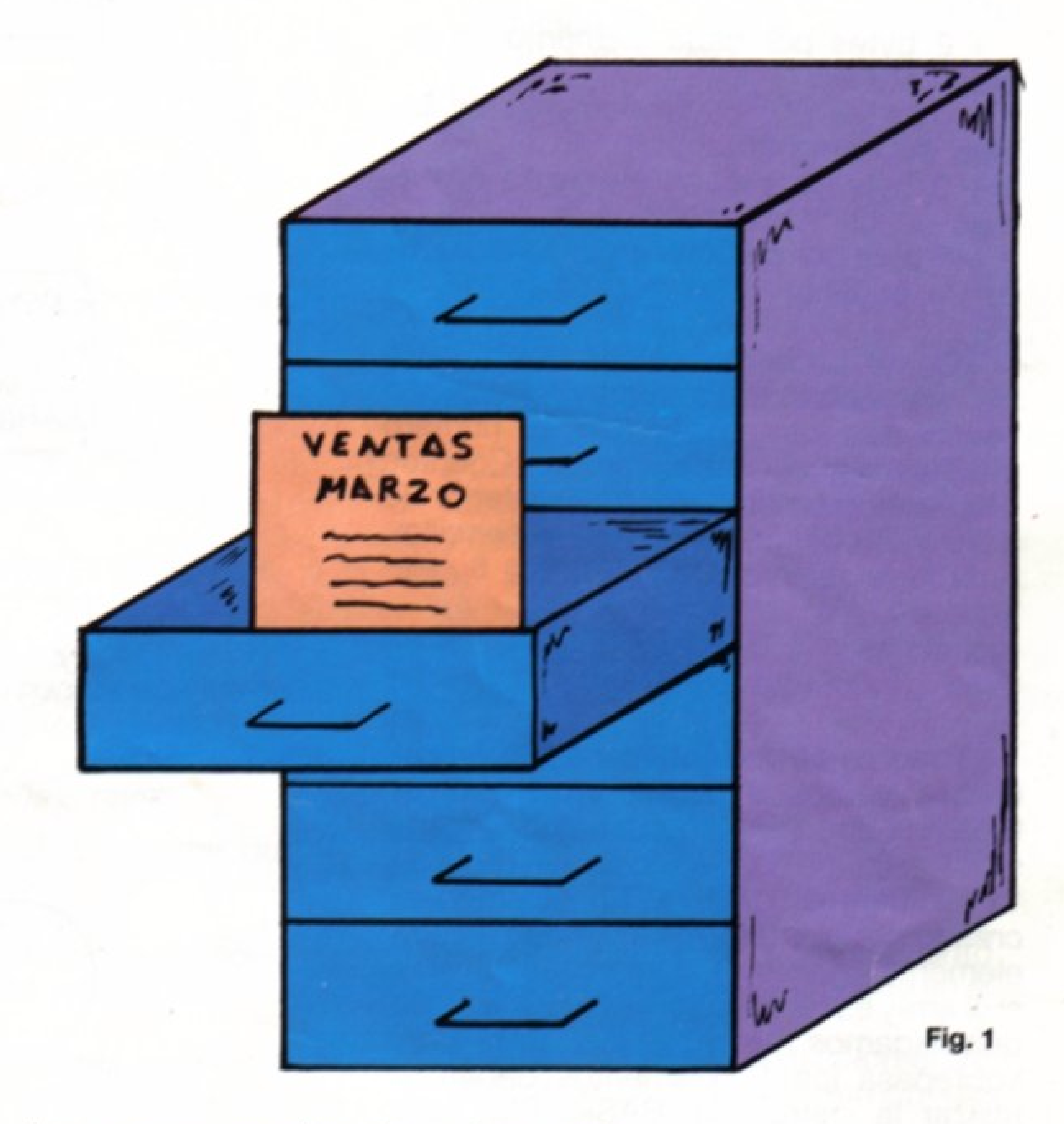


Fig. 1

Los arrays pueden tener una sola dimensión (como los que hemos visto hasta ahora) o varias dimensiones. Una tabla de dos dimensiones equivale a una matriz, es decir esta organizada también por filas y columnas. Así, si tenemos una array de dos dimensiones con tres filas y dos columnas haremos referencia a sus elementos, mediante el nombre del array y los índices —en este caso— dos separados por comas. Por ejemplo:
PP (2,1) Elemento del array PP fila 2 y columna 1.
Gráficamente quedará así:

COLUMNA

FILA		
	PP(2,1)	

¿Cuántas dimensiones y cuántos elementos por dimensión puede tener un array? Puede haber hasta 256 dimensiones y 32.767 elementos por dimensión. Este es el tamaño teórico, en la práctica esto viene limitado por el espacio disponible en memoria y/o el tamaño de la línea en memoria se puede calcular teniendo en cuenta lo siguiente:

5 bytes del nombre de la tabla.
+ 2 bytes por cada dimensión de la tabla.
+ 2 bytes por cada elemento en tablas enteras.
o + 5 bytes por cada elemento en tablas de coma flotante.
o + 3 bytes por cada elemento en tablas de cadena.
y + 1 byte por carácter en cada elemento de cadena.

¿Cómo podemos calcular el número de elementos de un array? Para ello tendremos que multiplicar el número de filas por el de columnas. ¿Cuántos elementos tendrá el array anterior (3 filas y 2 columnas)? Es muy sencillo:

Nº de elementos = 3 × 2 = 6 elementos
columnas
filas

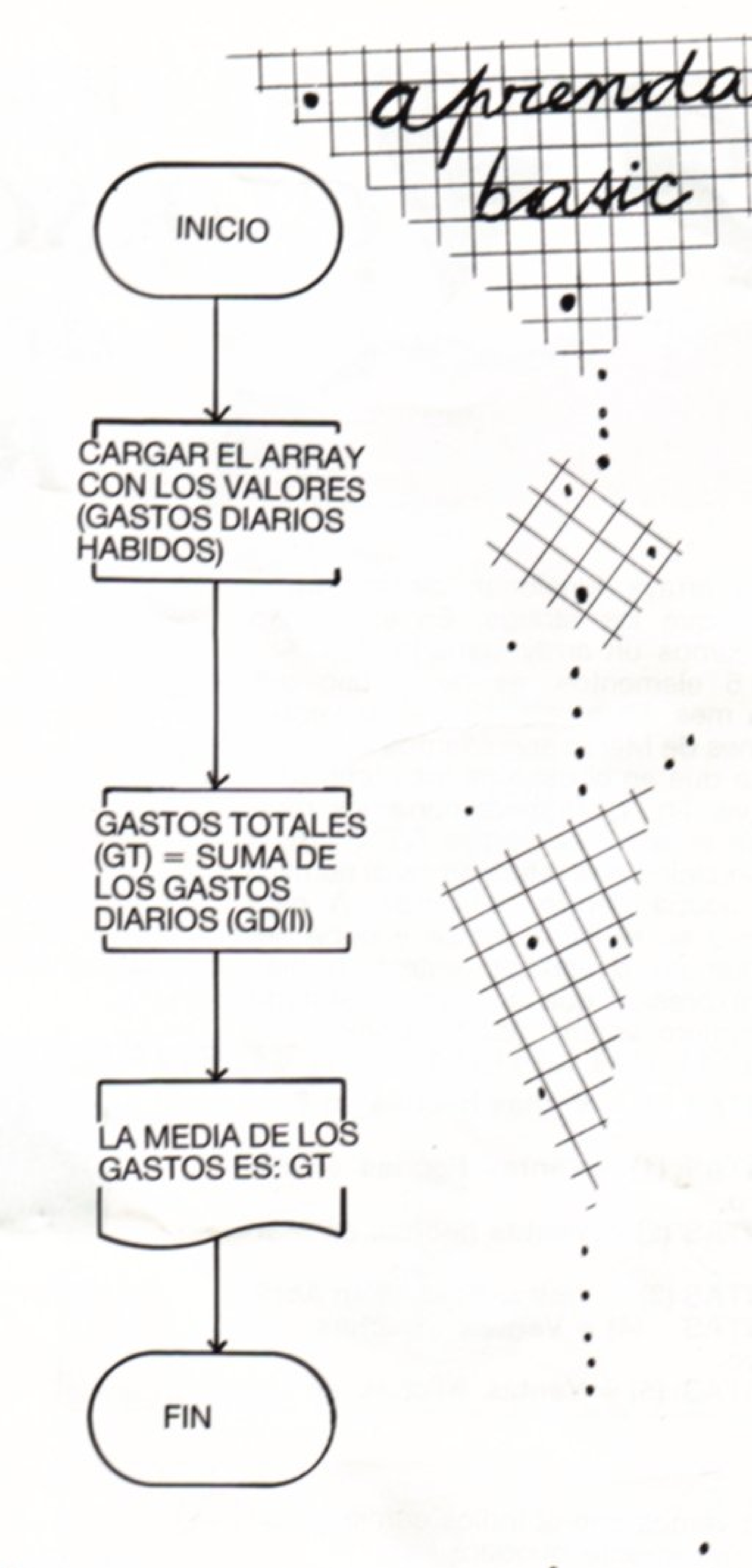
¿Tendremos que notificarle al intérprete que vamos a utilizar un array? Si tenemos una tabla con una sola dimensión y el índice no excede de 10 (Tabla de 11 elementos, de 0 a 10) el array es creado por el intérprete asignado a cada elemento un valor 0, o una cadena vacia si el array es de caracteres, la primera vez que hagamos referencia a él. Si el array sobrepasa los 11 elementos debemos utilizar la instrucción BASIC DIM (que estudiaremos más adelante) para definir el tipo y tamaño de la tabla.

Vamos a ver algunos ejemplos de nombres de tablas y de sus tipos:

A $ (0) = «HOLA»	Tabla de cadena
DI (I %) = 125.6	Tabla de coma flotante
SUM % (K % (1)) = 12	Tabla de enteros
MES $ (2) = «febrero»	Tabla de cadena

Para finalizar vamos a hacer un pequeño programita utilizando lo que sabemos hasta ahora. Queremos hallar los gastos semanales totales, teniendo como datos los habidos cada día.

Lo primero que debemos hacer es un organigrama, es decir, una representación gráfica de los pasos que debemos seguir. En este caso es muy sencillo:



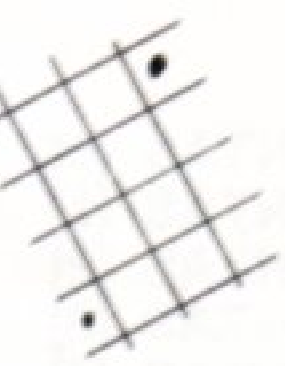


Fig. 2

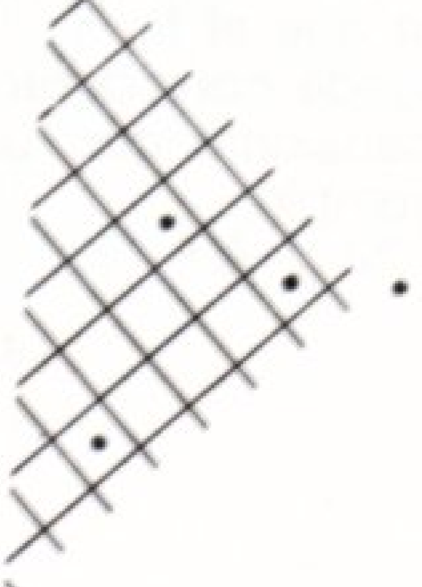
¿Qué variable vamos a utilizar?

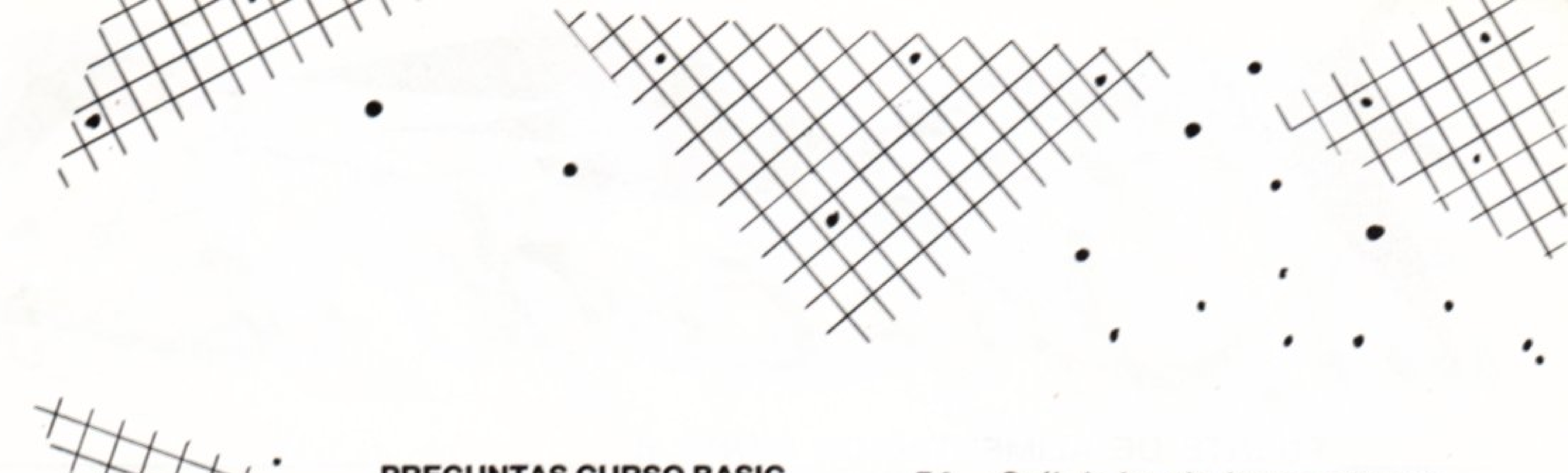
GT = Variable real donde vamos a almacenar los gastos totales.

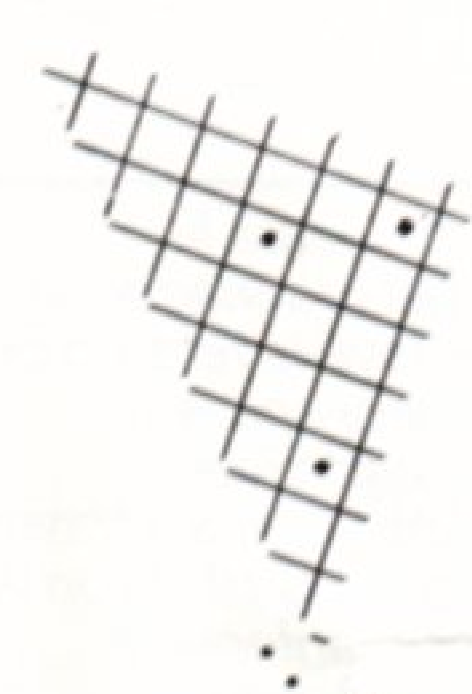
GD(I) = Array unidimensional, con 7 elementos - el índice varía de 0 a 6 - donde tenemos almacenados los gastos diarios. Así GD(0) = gastos habidos de lunes...

El programa quedará como sigue:

```
10 GD(0) = 345:GD(1) = 765:GD(2)
   = 999:GD(3) = 128:GD(4) = 140
20 GD(5) = 1567:GD(6) = 2500
30 GT = GD(0) + GD(1) + GD(2)
   + GD(3) = GD(4) = GD(5) + GD(6)
40 PRINT «LOS GASTOS TOTALES SE-
MANALES SON»; GT 50 END
```







PREGUNTAS CURSO BASIC

1.ª El índice de un array puede ser:

a) Cualquier número real.
b) Un número variable entera o una expresión cuyo resultado sea un entero
c) Una variable de cadena.
d) Ninguna de las anteriores.

2.ª El número máximo de elementos por dimensión es de 62767 por que:

a) Hay que poner un límite.
b) El índice es un entero.
c) No caben más en memoria.
d) Ninguna de las anteriores.

3.ª El índice de un array comienza en:

a) 1
b) 0
c) Depende del tipo de array.
d) Ninguna de las anteriores.

4.ª Los nombres de los arrays:

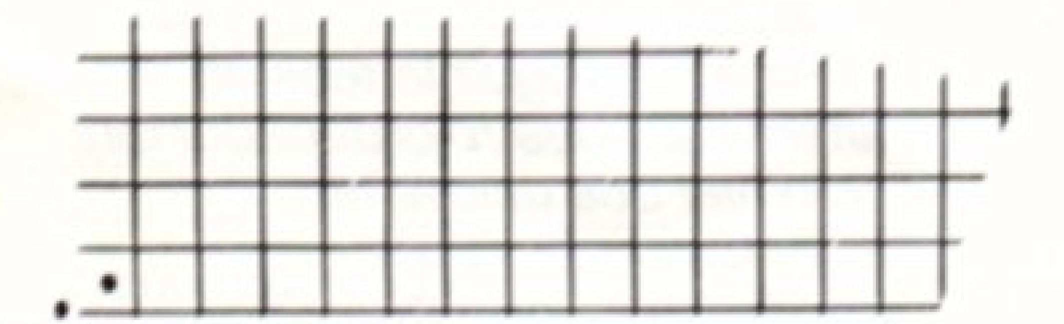
a) Siguen las mismas normas que los de las variables.
b) Varia el identificador.
c) No hay ninguna norma especial.
d) Ninguna de las anteriores.

5.ª ¿Cuál de las siguientes asignaciones de tabla de cadena son correctas?

a) A $ (10) = «HOLA».
b) B % (2) = «LUNES».
c) SN $ (0) = «SI».
d) a) y c)

6.ª ¿Cuál de las siguientes asignaciones de tablas enteras son correctas?

a) G % (1,2,3) = 5
b) D(4) = 2.5
c) DI(6,4) = «6»
d) Ninguna de las anteriores.



7.ª ¿Cuál de las siguientes asignaciones de tablas reales es correcta?

a) HOLA (5) = 185.4
b) SI (1,3) = «NO»
c) NO$ (3) = 4.8
d) Ninguna de las anteriores.

8.ª El número máximo de dimensiones es de:

a) 128
b) 32
c) 256
d) Ninguna de las anteriores.

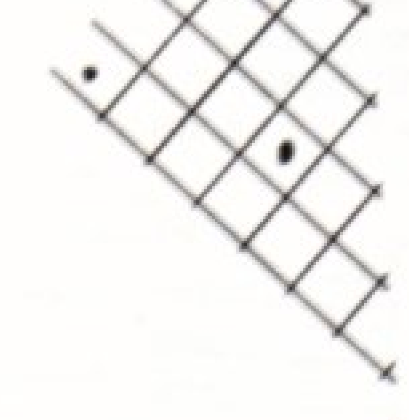
9.ª ¿Cuántos bytes de memoria ocuparía una tabla de enteros con tres filas y tres columnas?
a) 27
b) 9
c) 14
d) Ninguna de las anteriores.

10.ª ¿Cuál de las siguientes afirmaciones es correcta?

a) Los elementos del array deben ser del tipo especificado en el nombre del mismo.
b) Solamente el primer elemento debe ser del tipo especificado en el nombre del array.
c) No hay ninguna relación entre el tipo de los datos y el nombre del array.
d) Ninguna de las anteriores.

RESPUESTAS CURSO BASIC

1.ª b)
2.ª b)
3.ª b)
4.ª a)
5.ª d)
6.ª a)
7.ª a)
8.ª c)
9.ª a)
10.ª a)



...Bricodore

FUENTE DE ALIMENTACION PARTE II

La fuente ininterrumpida a diseñar tiene dos versiones; una aprovecha la fuente del propio Commodore y la otra no. En ambos casos conseguiremos tres alimentaciones aparte de las necesarias para el Commodore, que son: + 12, −12 y + 5 voltios, que para muchas aplicaciones posteriores, os serán necesarias. El diseño de esta fuente hace que aunque falle la tensión del ordenador, esté alimentado con una batería de + 12 voltios, que mediante el regulador 7805 mantendrá el programa en memoria hasta que vuelva la tensión.

Para los dos montajes necesitaremos un transformador de entrada 220 v y 12 + 12 voltios de salida. Su potencial nominal será de 5 a 6 VA y para el montaje en el cual no usamos la fuente de ordenador, además necesitaremos otro de 220 voltios de entrada y 9 voltios de salida, debiéndonos proporcionar un amperio de corriente.

La primera parte de ambos montajes es igual, tan solo cambia el valor de un condensador, el cual ya os indicaremos.

Lo primero que haremos será conseguir el rectificado necesario para pasar de corriente alterna a continua. Para ello utilizaremos un puente rectificador de 40 voltios y 5 amperios. Las dos salidas del secundario del transformador se conectarán a las tomas del puente rectificador marcadas con el signo ~ y la toma central del transformador a la parte común de nuestro circuito (masa).

La tensión rectificada de esta forma, nos dará una salida de doble onda, tanto en el positivo como en el negativo, se filtrarán mediante dos condensadores electrolíticos; uno de 1000 microfaradios para el esquema 1 y 2200 microfaradios para el esquema 2 a 30 voltios para la tensión positiva, y en ambos esquemas un condensador de 470 microfaradios a 30 voltios para la tensión negativa.

A las salidas del filtro conectaremos unos circuitos reguladores de tensión para cada una de las salidas; para la de + 12 voltios usaremos un 7812 y un 7912 para la de −12 voltios. A cada uno de estos reguladores les pondremos unos disipadores de calor atornillados a ellos. Como disipadores podemos utilizar una placa de aluminio de 12 cm^2 de superficie y 2 mm. de grosor. También podéis comprar un disipador equivalente en las tiendas de electónica.

A las salidas de todos los reguladores, deberéis conectar, como véis en los esquemas, un condensador de poliester de 110 K para conseguir una tensión de salida estable.

A partir de ahora ambos esquemas se diferenciarán en el conexionado y algún componente.

ESQUEMA 1. Para la alimentación del ordenador de +5 voltios, se lleva la tensión de salida del circuito, una vez filtrado y rectificado (ver esquema), a una bateria a través de una resistencia de 10 K y un W cuya misión es mantener la bateria cargada. A esta resistencia se conectará un regulador del tipo 7805 para conseguir una alimentación de + 5 voltios. Este regulador deberá tener un disipador de aletas con una base mínima de disipador de 60 cm^2 y un diodo con su pata común del tipo BY255, que hará que entre en funcionamiento al fallar la alimentación principal.

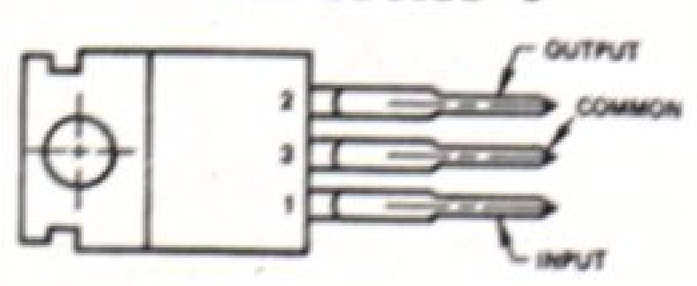
La salida de este circuito, se

ESQUEMA 2. Al contrario que en el esquema 1, que necesitamos la propia alimentación del ordenador, como fuente principal y si falta ésta que nos entre la de reserva. En este montaje no se necesita esta alimentación.

Para ello llevaremos un hilo desde el puente rectificador a un diodo tipo BY255 y de éste a la entrada del regulador. También llevaremos otro hilo a una resistencia de 10 K 1 W para que nos cargue la batería, como indicamos antes, y del positivo de ésta a través de un diodo BY255 a la entrada del regulador (ver esquema 2). Este debe ser del tipo 7805, con un disipador de 60 cm^2. La salida del regulador la conectaremos al pin 4 de un conector DIN de 360° y la masa de nuestro circuito al pin 2 de este conector.

Para conseguir la tensión alterna de nuestro ordenador tan solo hay que utilizar el transformador T2 ya

CONNECTION DIAGRAMS
TO-220 PACKAGE
(TOP VIEW)
PACKAGE OUTLINE GH
PACKAGE CODE U



ORDER INFORMATION

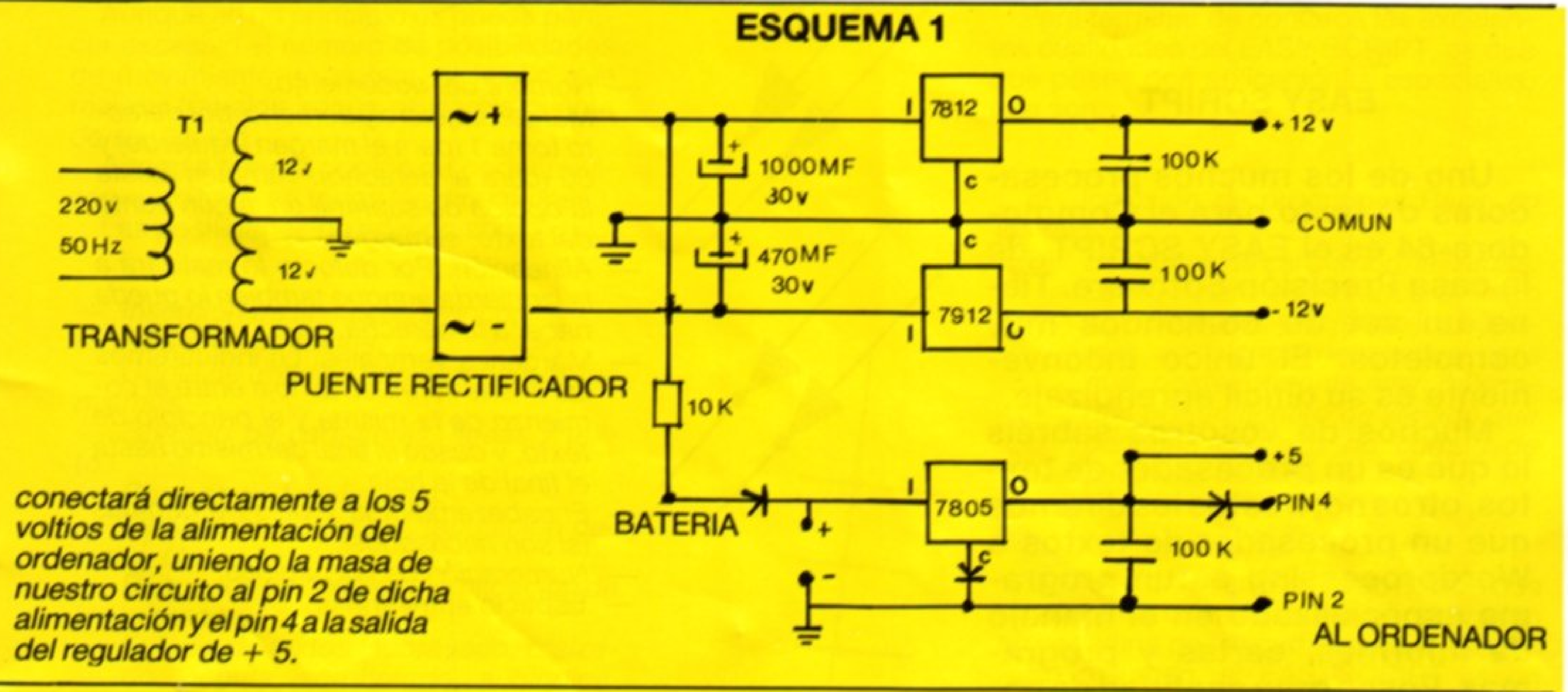
OUTPUT VOLTAGE	TYPE	PART NO.
5 V	µA7805C	µA7805UC
6 V	µA7806C	µA7806UC
8 V	µA7808C	µA7808UC
8.5 V	µA7885C	µA7885UC
12 V	µA7812C	µA7812UC
15 V	µA7815C	µA7815UC
18 V	µA7818C	µA7818UC
24 V	µA7824C	µA7824UC

CONECTOR DE ALIM.

+5V
MASA
9V∿
9V∿



ESQUEMA 1



conectará directamente a los 5 voltios de la alimentación del ordenador, uniendo la masa de nuestro circuito al pin 2 de dicha alimentación y el pin 4 a la salida del regulador de + 5.

ESQUEMA 2

T1
220 v
50 Hz
TRANSFORMADOR
PUENTE RECTIFICADOR
+ 2 200 MF
30 v
470 MF
30 v
7812
7912
7805
100 K
100 K
100 K
10 K
BATERIA
+ 12
COMUN
- 12
+ 5
PIN 4
PIN 2
AL ORDENADOR
T2
220 v
50 Hz
TRANSFORMADOR
9 v -
PIN 7
9 v -
PIN 8

especificado al principio, conectando las salidad del secundario, una al pin 7 y otra al pin 8 del conector de 360º. Con esto ya dispondremos de nuestra alimentación de 9 voltios alterna.

Esperamos que los montajes os resulten sencillos.



EASY SCRIPT

Uno de los muchos procesadores de texto para el Commodore-64 es el EASY SCRIPT, de la casa Precisión Software. Tiene un set de comandos muy completos. El único inconveniente es su difícil aprendizaje.

Muchos de vosotros sabréis lo que es un procesador de textos, otros no; a estos les diremos que un procesador de textos o Wordprocessing es un programa especializado en el manejo de informes, cartas y programas. Posee gran cantidad de comandos para la edición, impresión, creación y almacenamiento de programas.





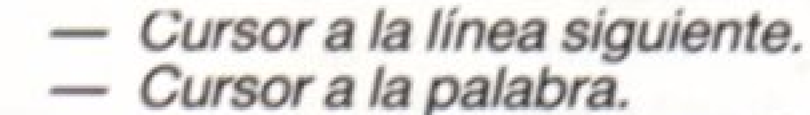
Para decidiros por el uso de un procesador u otro debéis mirar además de los comandos de creación y almacenamiento, los de edición e impresión; estos son precisamente los que pueden hacer que un procesador de textos sea útil o no.

Por lo expuesto anteriormente podemos afirmar que el EASY SCRIPT tiene excelentes condiciones para ser uno de los mejores procesadores del Commodore 64, ya que tiene gran número de comandos para la edición e impresión de textos, aunque a veces el recordarlos sea una ardua tarea para el usuario que sólo lo utiliza de vez en cuando.

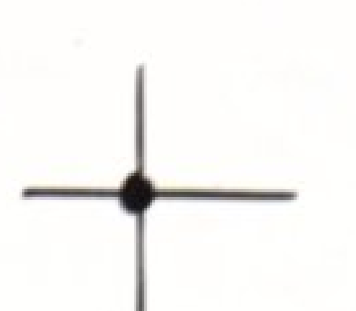
Siempre que creemos un documento con un procesador deberemos indicarle una serie de características a seguir página a página. Estas características se resumen en el EASYSCRIPT en:

— *Nombre del documento.*
— *Margen izquiero y derecho; por defecto toma 1 (para el margen izquierdo) y 80 (para el derecho). También existe la opción de suprimir en, algún punto del texto, el margen.*
— *Alineación. Por defecto lo realizará a la izquierda aunque también lo pueda hacer a la derecha.*
— Márgenes verticales. Le indicaremos *las líneas libres de la hoja entre el comienzo de la misma y el principio de texto, y desde el final del mismo hasta el final de la hoja.*
— *Encabezamientos y pies de páginas (si son necesarios).*
— *Numeración de las páginas.*
— Espacio entre líneas.

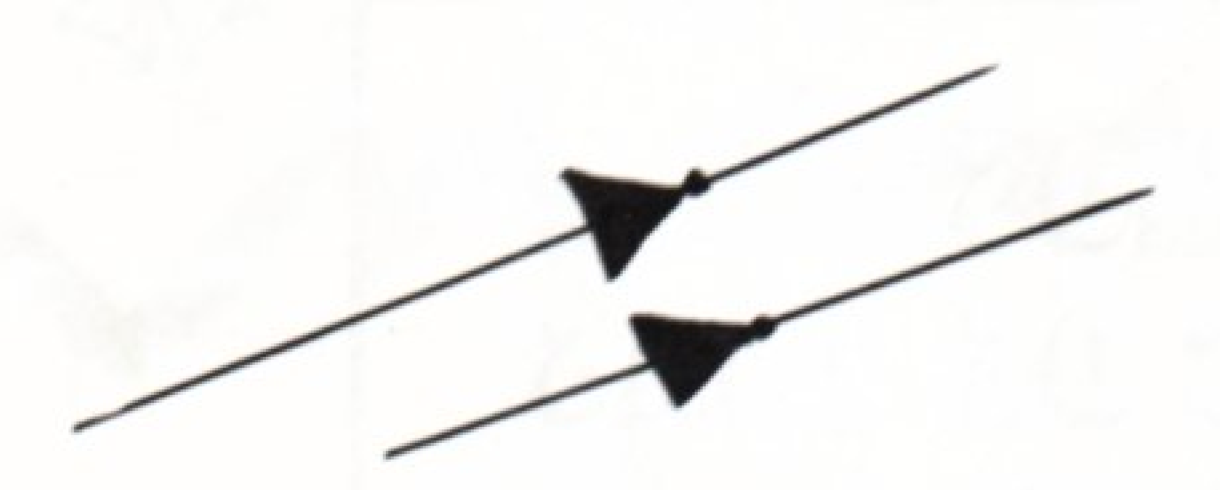
Una de las características fundamentales que debe tener un procesador de textos es la de poder mover el cursor fácilmente para realizar las modificaciones que sean necesarias; pero como un documento puede tener muchísimas páginas y podemos necesitar modificar muchas de ellas, un buen procesador además de tener movimientos de cursor de carácter a carácter en horizontal o de línea a línea en vertical, debe tener otros más. El EASY SCRIPT dispone, además de los movimientos normales de cursor, de los siguientes:



— *Cursor a la línea siguiente.*
— *Cursor a la palabra.*
— *Cursor a la última línea de la palabra anterior.*
— *Cursor a la última posición de la línea anterior.*
— *Cursor al inicio de texto.*
— *Cursor a la primera posición de la pantalla.*
— *Cursor al final del texto.*
— *Cursor a una línea determinada.*
— *Cursor a la última línea escrita.*
— *Cursor a la próxima pantalla.*
— *Cursor a la pantalla anterior.*







Aunque en un principio os pueda parecer excesivo el número de posibilidades de movimiento de cursor, si tenéis que realizar trabajos largos os daréis cuenta de su utilidad.

Además tiene la opción de poder realizar «scrolling» en todas las direcciones.

Las posilidades de edición son:

— *Inserción de líneas en blanco.*
— *Inserción de textos.*
— *Copia de textos y transferencia de los mismos.*
— *Borrado de caracteres, líneas y textos.*
— *Borrado de tabuladores.*
— *Tabulación de textos, tabulación decimal, etc.*
— *Búsqueda de textos y modificación de los mismos.*
— *Enfasis de textos (subrayado, realzado, negrita, sombreado, subindice, supraindice).*

También permite la generación de caracteres especiales o secuencia de «ESC» para la impresora.

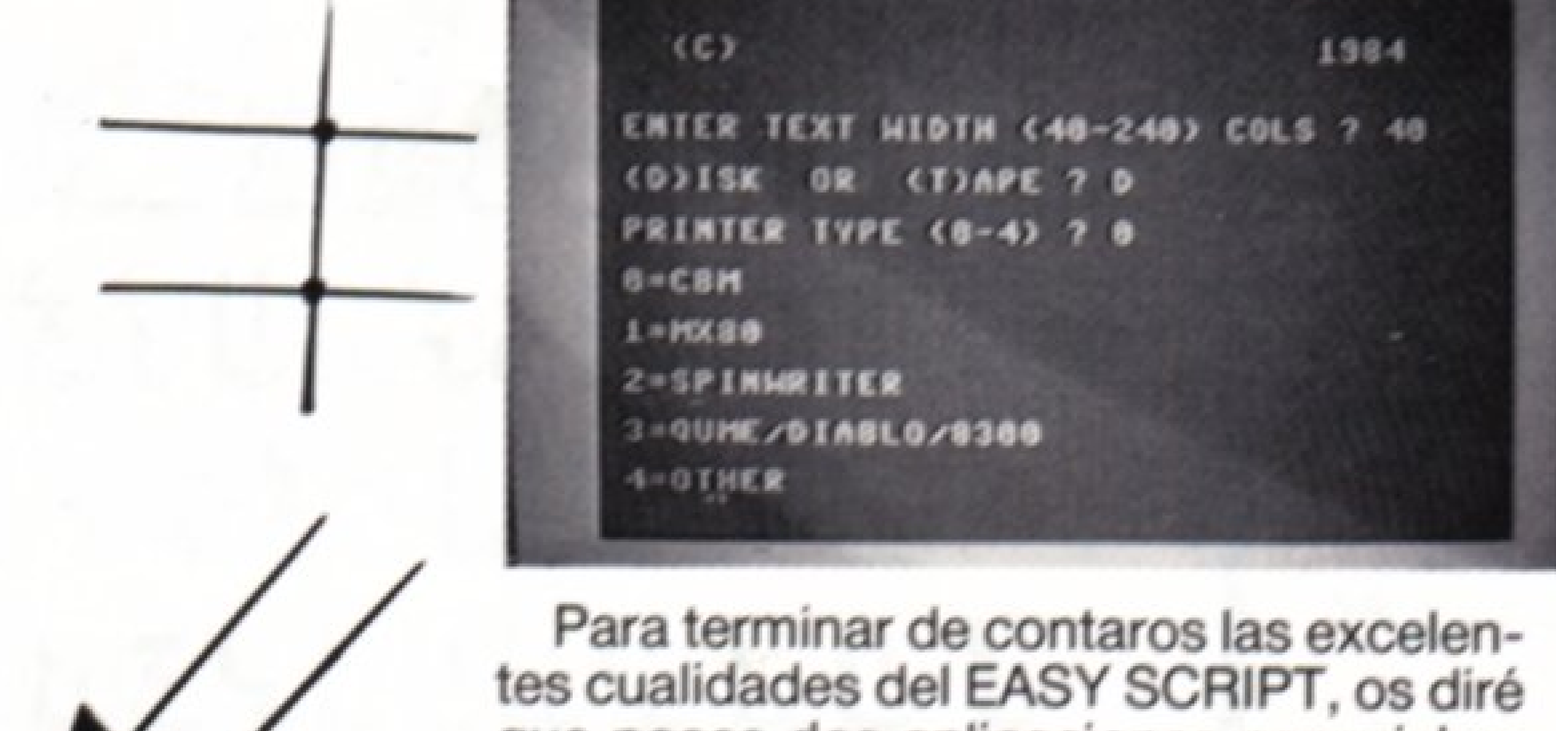
Otra opción muy interesante del EASY SCRIPT es la de encadenar documentos. La utilizaremos cuando tengamos que fragmentar un documento, debido a su excesiva longitud.

Las características de impresión de este procesador, aparte de las comunes a todos, son:

— *Líneas por pulgada.*
— *Punto y aparte en vez de Return al final de línea.*
— *Backspace.*
— *Caracteres por pulgada.*

Los comandos de disco son todos los existentes.

Posee también unas tablas de códigos de error, que nos indicarán cual ha sido la posible causa de que se haya producido un fallo.



Para terminar de contaros las excelentes cualidades del EASY SCRIPT, os diré que posee dos aplicaciones especiales, que son:

a) Inclusión de programas Basic en los textos.

b) Escrituras de cartas normalizadas. Esto se hará escribiendo el texto común a ellas y dejando los espacios en blanco necesarios para que, posteriormente sean rellenados por un fichero de llenado que contiene el texto no común de cada una de ellas.

Un fichero de llenado, es un fichero que contiene los bloques de textos que varían en cada una de las cartas normalizadas.

En resumen, el EASYSCRIPT es un magnífico procesador de textos aunque su aprendizaje sea algo laborioso.





Uno de los grandes inconvenientes del C-64 es la relativa baja velocidad de su unidad de discos, la unidad 1541.

A raiz de esto, han salido al mercado una serie de programas y accesorios hardware para intentar subsanar esta deficiencia, como el 1541 ¡FLASH! de la compañía Skyles Electric Works motivo de este artículo.

Programa	NORMAL	FLASH!
Superbase 64	02m05s	00m58s
Easy Script	01m02s	00m31s
Logo	01m53s	00m52s
Ultrabasic	00m33s	00m11s
Kawasaki Rythm Rocker	01m42s	00m32s
Basic Lightning	01m03s	00m23s
Bruce Lee	03m06s	01m24s

Su aumento de velocidad es, en carga, de tres veces la velocidad normal, así por ejemplo un programa que tarde 1.5 minutos en cargarse a velocidad normal, con el ¡FLASH! tardará 1/2 minutos en completar la carga. Hay, no obstante, algunas excepciones a esto, casi siempre ocasionadas por las protecciones cada vez más sostificadas del software, que incluso llegan a hacer incompatible el !FLASH! con la carga de estos programas; aunque realmente solo nos hemos encontrado con un mínimo de programas protegidos que no funcionen con él.

A continuación se relaciona una tabla de velocidades de carga de algunos programas comerciales con sus tiempos respectivos.

La velocidad de grabación es, desafortunadamente, prácticamente la misma que la normal, habiendo un pequeño aumento de velocidad pero que no es demasiado apreciable.

El ¡FLASH! se presenta como dos Eprom, montadas cada una en su zócalo soldado a una pequeña placa de circuito impreso, una de ellas sustituyendo al sistema operativo de la unidad de discos, y otra sustituyendo a la RON del KERNAL en el C-64. Otra placa de circuito impreso conectase al port del usuario. Esta lleva el conector necesario para su conexión en un extremo y en el otro un duplicado del port del usuario para así no perder las facilidades de esta comunicación con el exterior. Como ejemplo es de destacar que no hemos observado ninguna incompatibilidad en la conexión de una impresora con interface tipo paralelo centronics y conectada al ordenador por medio del cable y software necesario al port del usuario. En la misma placa hay por debajo dos pequeños enchufes, uno sirve para la conexión directa con la VIA (6522) de la unidad de discos y el otro per-

mite, por medio de un pequeño interruptor, conmutador la RON del KERNAL con la EPRON del ¡FLASH! y quedar asi en las condiciones inicales del ordenador o sea, con la velocidad de origen de la 1541. Además de lo anteriormente expuesto, el sistema operativo añade una serie de comandos para facilitar el manejo de la unidad de discos, que son practicamente los mismos que los que tiene el programa DOS 5.1 en el disco demostración de la 1541. Para disponer de ellos solo es necesario hacer SYS 65526. A partir de ese momento con solo teclear el símbolo AT ingles, seguido del comando de disco se efectuará la misma operación que si se escribiese OPEN 15, 8, 15 «(comando) O»: close 15, y para leer el canal de error solo es necesario pulsar la teclar del AT ingles y return dandonos como respuesta el estado del mismo.

Otra de las ventajas del ¡FLASH! es su velocidad de formateo que pasa de 1 minuto 20 segundos, a la velocidad normal, a 18 segundos, con el consiguiente ahorro de tiempo que ello supone.

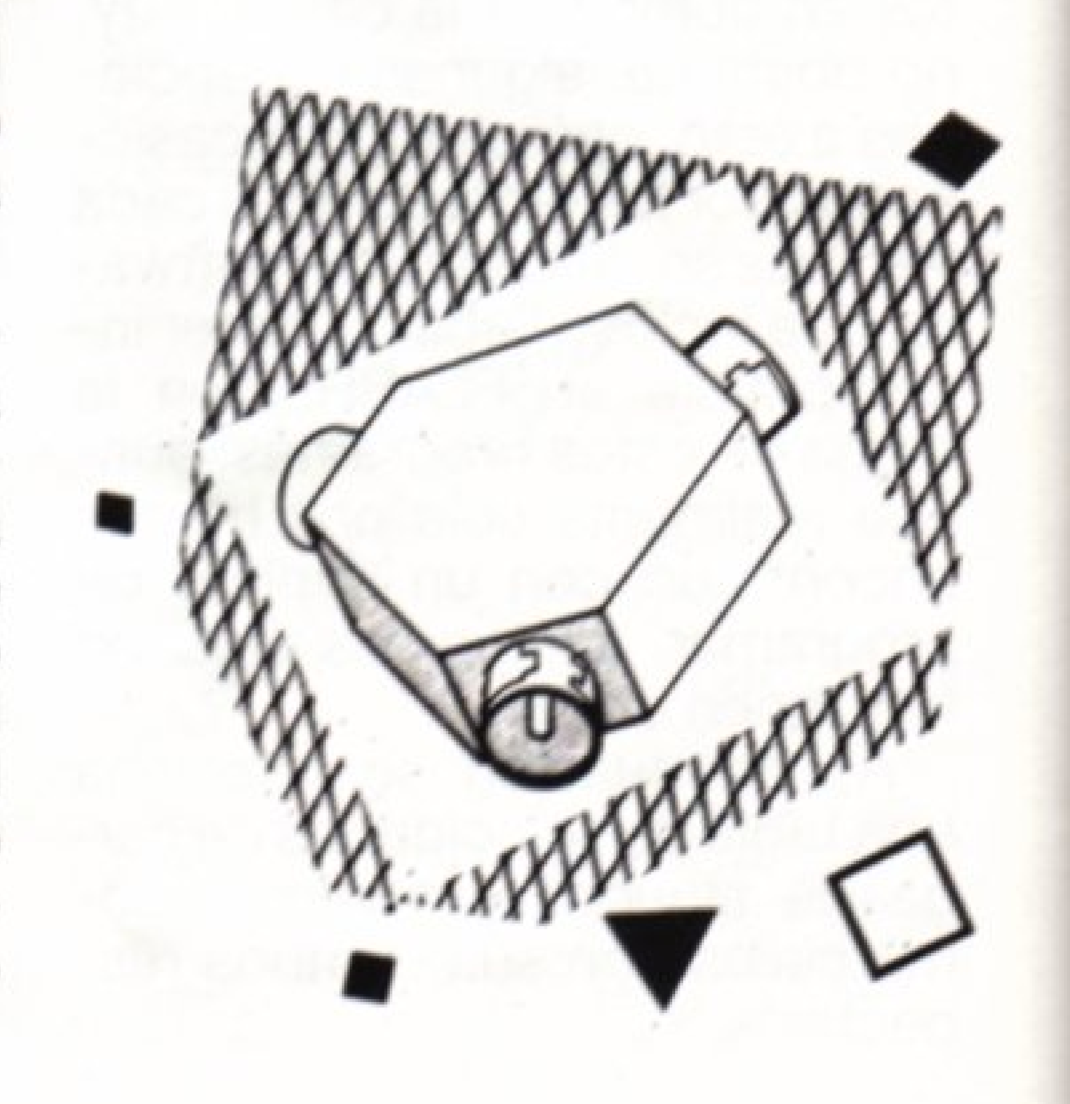
Existen también mejoras en el editor de pantalla completa del ordenador. Así por ejemplo pulsando al tecla CTRL y —, el cursor se desplaza al pie de la pantalla. La tecla commodore y el sínbolo de «elevado a» hace salir del modo comillas, tan incómodo algunas veces. El SHIFT izquierdo o el SHIFT/LOCK detiene totalmente el desplazamiento hacia arriba de la pantalla en los listados o impresiones largas. La Tecla Commodore y CLR/HOME borran la pantalla desde donde esté el cursor hacia abajo etc.

Bajo nuestro punto de vista el único inconveniente que tiene el ¡FLASH! es que no soporta el comando SAVE para el cassette, lo cual puede constituir, algunas veces, un grave problema; aunque siempre queda el recurso de grabar lo que estamos haciendo en ese momento en disco y desconectar el ¡FLASH!.

La carga de programas desde cassette funciona perfectamente, la única diferencia es que como el ¡FLASH! considera dispositivo principal de salida el disco, para cargar de cinta es necesario poner LOAD «nombre», 1; y sin embargo para cargar de disco un programa normal basta poner LOAD «nombre» (sin ni siquiera cerrar comillas); lo mismo que para SAVE a disco. Con respecto a esto hay que señalar que la secuencia SHIFT/RUN STOP no activa la carga del cassette con ejecución automática, sino que provoca LOAD «:*», 8, 1 que nos carga el programa que esté en la cabecera del directorio. Esto puede ser un inconveniente para la carga de programas comerciales en cinta que utilicen en su esquema de protección el CHR$ (131), que es el código ASCII en el C-64 de la secuencia SHIFT/RUN STOP.

Hay también otra serie de comandos añadidos al sistema operativo de disco que comienzan por la letra Z y sirven para programar la transferencia de datos incluso más deprisa de lo que ya lo hace en carga normal (el rango de ajuste de velocidad es de 0 a 15 estando por defecto en 1), pero siendo esto una característica solo del ¡FLASH! y no siendo compatible con una 1541 normal valga el nombrarlo sin más. No obstante hay que destacar que Z8S (low) pasa a velocidad 1541 y Z8F (ast) pasa a velocidad ¡FLASH!.

Solo resta decir que su instalación es bastante sencilla consiste únicamente en levantar las ROM de su correspondiente zócalo, sustituirlas por las EPROM, conectar la placa del circuito impreso en el port del usuario, levantar uno de los 6522 de la unidad de discos y hacer una pequeña conexión. El manual, en cuanto a la instalación, es muy claro, y tiene numerosos dibujos y fotografías aclaratorias, no siendo necesario en ningún momento más que un par de destornilladores. A lo anterior hay que hacer una pequeña salvedad; debido a que algunos C-64 llevan sus circuitos integrados soldados a la placa de circuito impreso, es necesario desoldar la ROM del KERNAL y poner un zócalo para poder hacer la instalación (esto no es recomendable hacerlo sino se dispone del material necesario).







PROGRAMAS BASIC

OESTE

El juego de este programa listado consiste en el duelo a muerte entre dos conocidos pistoleros en la calle principal de Dodge City. El de la derecha, llamado por sus secuaces «Mac Manolenta Kid», es uno de los revólveres más rápidos del Sur de Pekos (se maneja con el joy-2), y el de la izquierda es el conocido pistolero y bandido «Joe el Sanguinario» (se maneja con el joy-1 o con las teclas Q-subir, A-bajar, W-disparo). Los dos se enfrentan para conocer cuál es el más rápido.

```
1 REM ********** OESTE ************
2 REM ****** JAVIER LOPEZ *********
3 REM ****************************
5 PRINT"":POKE53280,0:POKE53281,13:POKE650,128
10 PRINT"                    *** ESPERE  UN  MOMENTO ***"
12 GOTO5000
15 PRINT"":FORA=0TO30:POKE1024+A,8:POKE55296+A,13:NEXT
17 FORA=800TO990:POKE1024+A,8:POKE55296+A,13:NEXT
20 PRINT"                                        "
23 PRINT"                                        "
25 PRINT"   OOO                OOS    11         "
28 PRINT"                     P                  "
30 PRINT"                             OOO        "
32 PRINT"                                        "
33 PRINT"                                        "
35 PRINT"   EL JUEGO CONSISTE                    "
38 PRINT"   EN MATAR AL VAQUERO                  "
40 PRINT"   CONTRARIO ANTES DE                   "
42 PRINT"   QUE SE AGOTE EL CAR                  "
43 PRINT"   GADOR.PERO DEBES TE                  "
45 PRINT"   NER EN CUENTA QUE                    "
48 PRINT"   NO PUEDE CHOCAR NI                   "
50 PRINT"   CON LA CARRETA NI                    "
53 PRINT"   CON LOS CACTUS.     P                "
55 PRINT"   R                   T     OESTE      "
58 PRINT"                                        "
60 PRINT"                                        "
70 GOSUB4100
72 CO=1:FA=20:GOSUB4500
80 PRINT"  IIIIIII              IIIIII"
85 POKE2040,216:POKE2041,217:POKE2042,218:POKE2043,221:POKE2044,221
90 POKEV+39,6:POKEV+40,0:POKEV+41,15:POKEV+42,0:POKEV+43,0
95 Y=100:Y1=100:ID=55:DI1=175:AV=0:VA=0:N1=802:N=817:OC=0:CHO=0
97 FORT=255TO0STEP-3.5
100 JW=PEEK(56321):JV=PEEK(56320)
105 IFJW=255THEN120
110 IFJW=254THENY=Y-2.2:GOTO130
115 IFJW=253THENY=Y+2.2:GOTO130
117 IFJW=239ANDVA<17ANDN1<808THENAL=Y:VA=17:N1=N1+1:GOTO130
120 GETA$:POKE198,0:IFA$="A"THENY=Y+2.2:GOTO130
122 IFA$="Q"THENY=Y-2.2:GOTO130
125 IFA$="W"ANDVA<17ANDN1<808THENAL=Y:VA=17:N1=N1+1:GOTO130
130 IFJV=127THEN152
135 IFJV=126THENY1=Y1-2.2:GOTO152
140 IFJV=125THENY1=Y1+2.2:GOTO152
145 IFJV=111ANDAV<17ANDN<823THENLA=Y1:AV=17:N=N+1
152 IFY<75THENY=75
154 IFY>170THENY=170
160 IFY1>170THENY1=170
163 IFY1<75THENY1=75
165 ID=ID+VA:DI1=DI1-AV
```

(Continúa pág. 19.)

cristal of zong

El juego que os presentamos, es muy ameno como podréis comprobar.

En él existen laberintos intercomunicados numerados del 1 al 9. En todos ellos existe una habitación cerrada, que podréis abrir siempre que encontréis la llave adecuada; está estará en algún lugar de los nueve laberintos. Con dicha llave tendréis acceso a uno de los nueve tesoros escondidos en las habitaciones. Estos pueden ser: unas botas mágicas, dos pociones contra monstruos, un quinque, una espada, un diamante, un arcón, una corona y un trono.

Cada uno de los nueve tesoros tiene cualidades distintas; por ejemplo, las botas os harán ir más deprisa, con las pociones podréis ser inmunes a los monstruos que os ataquen, etc. Entre estos objetos existen dos, que son más codiciados que los demás, son: la corona y el trono. Ambos os harán pasar directamente a otro nivel. También encontraréis antorchas que os alargarán la vida y espadas con las que podréis defenderos de los monstruos durante un tiempo. Como veis es bastante entretenido.

CONTROLES

Con el joystick conectado a la puerta dos controlarás el juego. Si das al ESPACIO el juego se detendrá.

PANTALLA

— En la parte superior de la pantalla aparecerá:

SCORE	**—**	**Tanteo del jugador.**
MEN	**—**	**Número de hombres que os quedan (máximo cinco).**
LEVEL	**—**	**Nivel del juego.**
HI	**—**	**Máximo tanteo obtenido.**

— En la parte inferior veréis:

TREASURES	**— Tesoros conseguídos, numerados según el laberinto donde se encontraba.**
KEYS	**— Llaves encontradas numeradas del 1 al 9.**
ROOM	**— Laberinto donde os encontrais.**
TOURCH TIME	**— Tiempo que os queda de vida.**



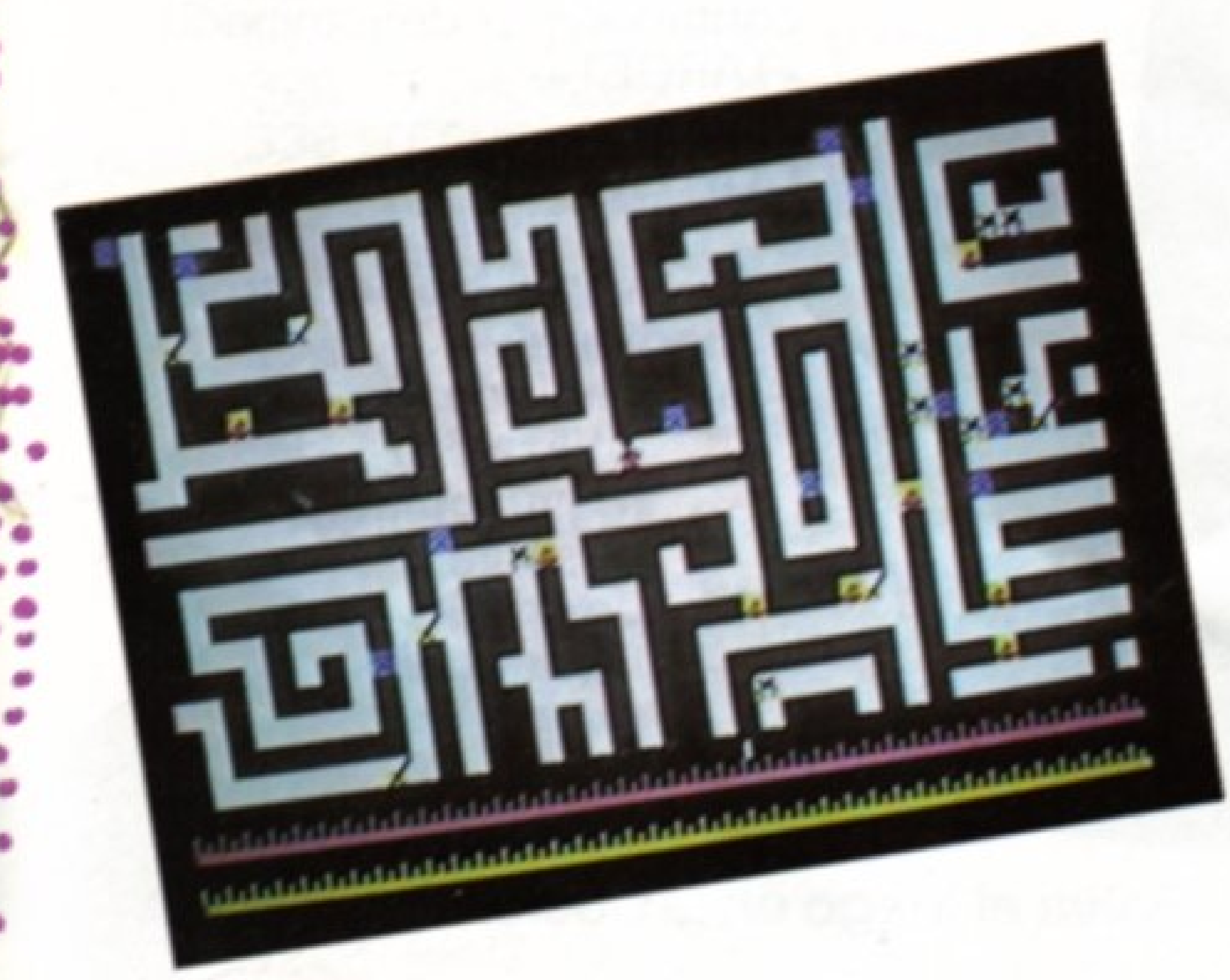


CONCLUSIONES

El juego es muy divertido y los monstruos, según aumentéis el nivel de juego serán distintos y cada vez más agresivos, os perseguirán de forma más persistente.



maziacs



Es un juego bastante entretenido aunque a veces se puede convertir en desesperante, pues cuando estás a punto de llegar a tu meta los MAZIACS (monstruos del laberinto) te asesinan brutalmente.

Existen tres niveles de dificultad. En el último de ellos podrás liberar a los encadenados una vez que te hallan facilitado información sobre donde se encuentra el tesoro: esto te hará el regreso más difícil al no poder volver a pedirles información sobre donde está tu guarida.



JUEGO

Consiste en llegar desde el lugar de inicio hasta donde se encuentra el tesoro y volver de nuevo, con el tesoro por supuesto al punto de partida (tu guarida). Para conseguirlo podrás pedir ayuda a unos personajes encadenados que irás encontrando por el camino. Ellos te indicarán por donde debes ir haciendo que cambie el color del camino que debes seguir. También puedes defenderte de los MAZIACS utilizando unas espadas que están guardadas en hornacinas en la pared, y aumentar vuestras reservas de energía comiendo de los alimentos que también encontraréisos en las hornacinas. Existen dos niveles en la parte inferior de la pantalla que te indicarán:

1) El nivel de energía que te queda.
2) La cantidad de ayudas que puedes seguir pidiendo.

CONTROLES

Existen dos formas de jugar:

a) Mediante teclado:
Q — Arriba.
A — Abajo.
Z — Izquierda.
X — Derecha.
V — Da una panorámica del laberinto.

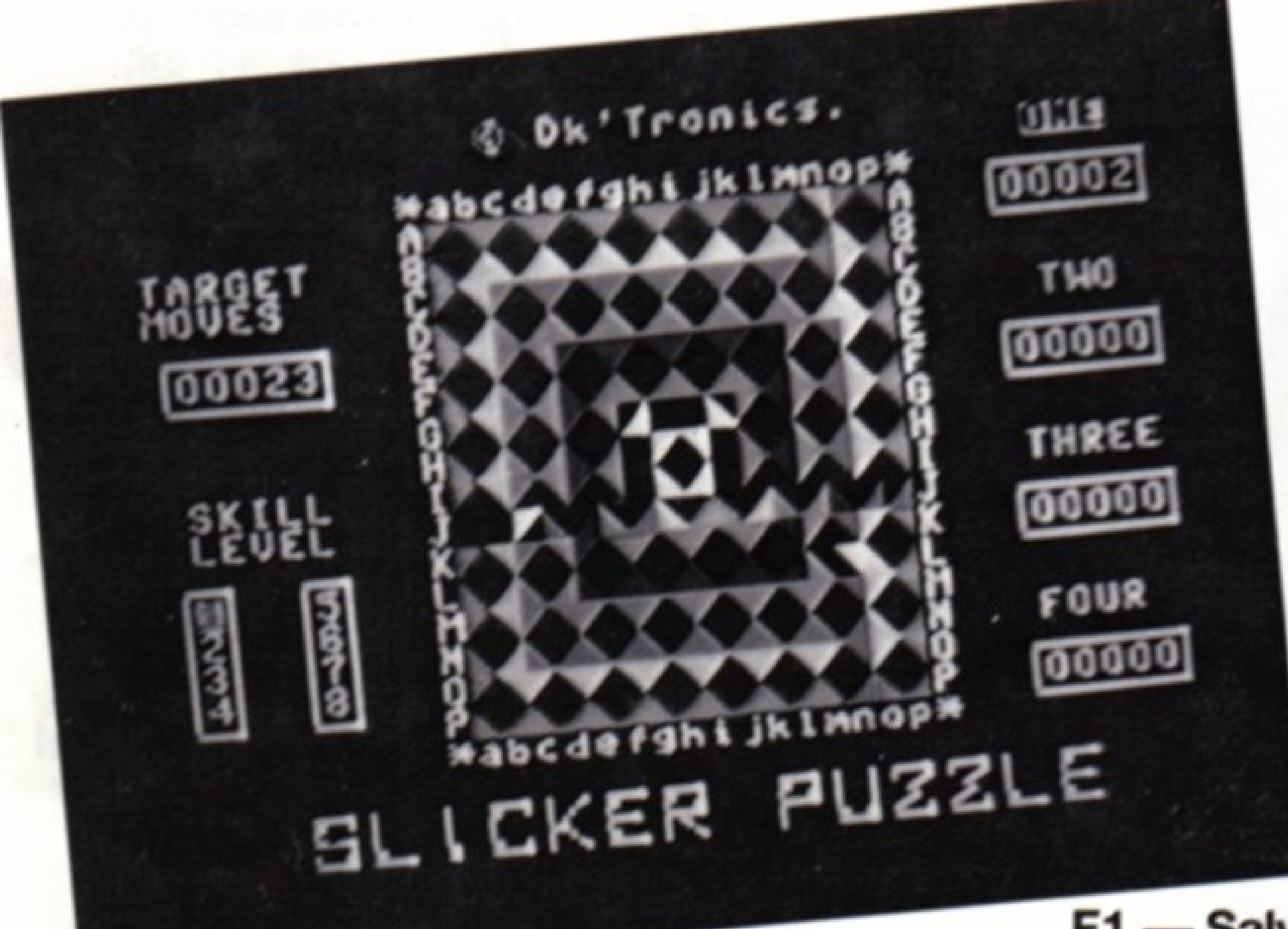
b) Mediante joystick:
Conectar el joy a la puerta dos. Apretando el botón de fuego lograrás ver el laberinto completo.

CONCLUSION

El MAZIACS aunque no sea de los mejores en cuanto a gráficos y sonido, pues es una adaptación de un juego del SPECTRUM, es bastante entretenido. Espero que no os desesperéis intentando llegar a vuestra meta en el laberito.



mosaico



Es un juego de DK'TRONICS que consiste en ordenar un mosaico en un número de movimientos inferior o igual al que indica un contador denominado «TARGET».

CONTROLES

El juego se controla mediante:

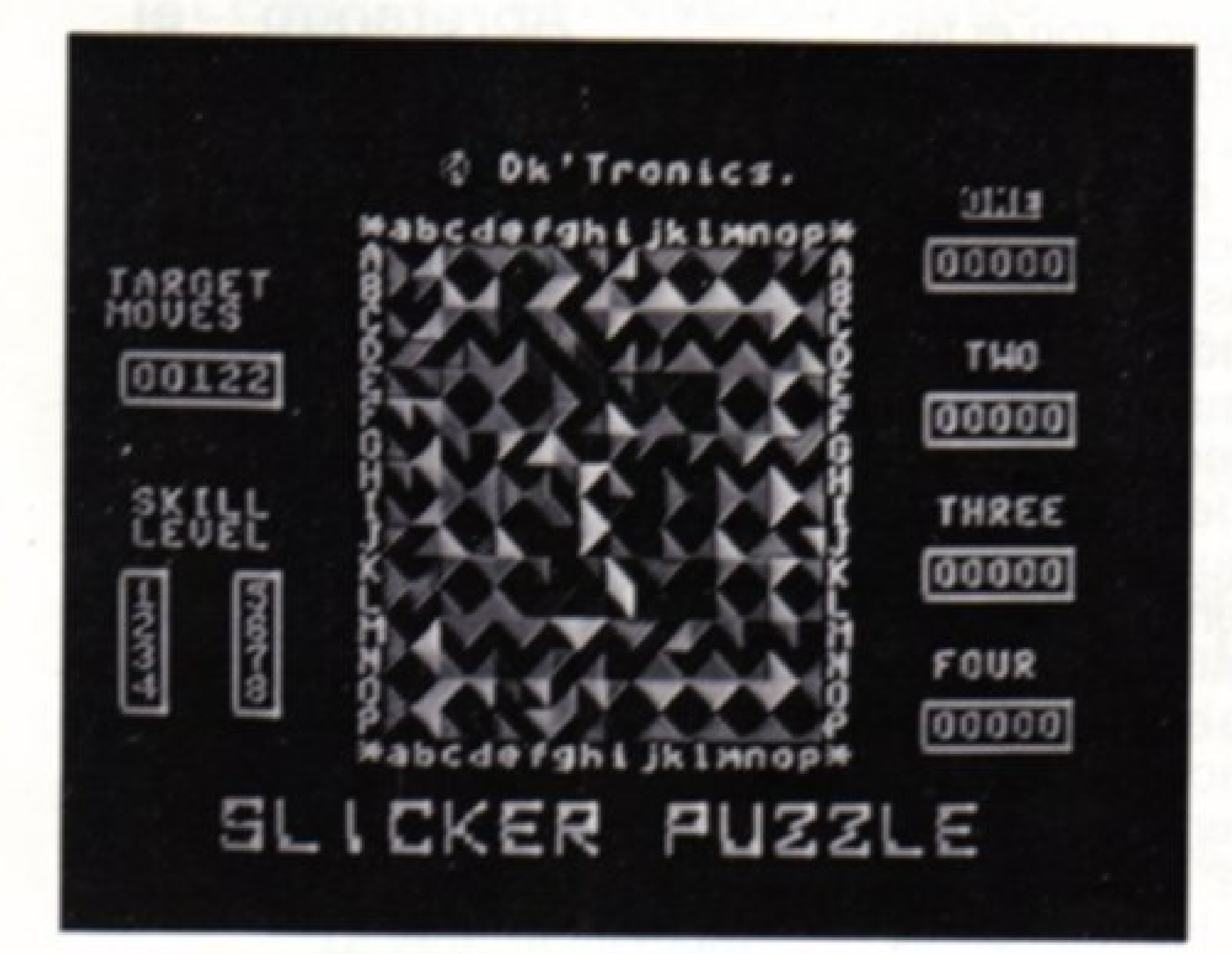
a) Teclado:
Utilizando letras mayúsculas y minúsculas.

b) Joystick:
Conectado a la puerta dos.

F1 — Salva el juego en la posición en la que esté.

F2 — Carga el juego de cinta en la posición en la que se encontraba.

ESPACIO — Visualiza el mosaico original.



JUEGO

Una vez cargado podréis escoger entre 8 niveles. Para elegir uno, sólo deberéis teclear el número correspondiente al nivel que deseeis. Par seleccionar el número de jugadores deberéis repetir la operación anterior. Una vez hecho esto el ordenador desordenará el mosaico.

Espero que os guste el juego.

Si lográis recomponer el mosaico en menos movimientos de los indicados por el TARGET, el ordenador os mostrará su enfado, pues no le gusta que le ganen, sacando en pantalla unos caracteres ininteligibles. Por el contrario si no lo conseguis os demostrará su alegría.

PROGRAMAS BASIC

(Viene de pág. 15.)

```
170 POKE1024+N1,8:POKE55296+N1,13
173 POKE1024+N,8:POKE55296+N,13
175 POKEV,55:POKEV+1,Y:POKEV+2,175:POKEV+3,Y1
180 POKEV+6,ID:POKEV+7,AL:POKEV+8,DI1:POKEV+9,LA
190 POKEV+27,7:POKEV+23,4:POKEV+4,107:POKEV+5,T:POKEV+29,4
200 IFID>195THENVA=0:ID=55:AL=0
210 IFDI1<40THENAV=0:DI1=175:LA=0
220 OC=PEEK(53278)
230 CHO=PEEK(53279)
235 IFOC=0THEN300
240 IFOC<>10THEN250
242 AL=0:FORTM=0TO300:NEXT:POKE2041,220
245 FA=10:CO=7:GOSUB4500:PRINT"■GANA EL AZUL":POKEV+6,0:POKEV+8,0
247 FORTY=0TO5000:NEXT:POKEV,0:POKEV+11,0:POKEV+13,0:GOTO420
250 IFOC<>17THEN260
252 LA=0:FORTM=0TO300:NEXT:POKE2040,219
255 FA=10:CO=7:GOSUB4500:PRINT"■GANA EL NEGRO":POKEV+6,0:POKEV+8,0
257 FORTY=0TO5000:NEXT:POKEV,0:POKEV+11,0:POKEV+13,0:GOTO420
260 IFOC=20THENLA=0:GOTO400
270 IFOC=12THENAL=0:GOTO400
300 REM RUTINA DE CHOQUE
310 P1=INT((Y-32)/8)+40:P3=INT(X-8)/8:P1=P1+P3:PX=P1+1024
320 IFCHO=20ORCHO=16THENLA=0
330 IFCHO=12ORCHO=42THENAL=0
340 IFN1=808ANDN=823ANDID=55ANDDI1=175THEN4050
400 NEXT
410 GOTO97
420 CO=4:FA=10:GOSUB4500:PRINT"                              ":GOTO72
4050 CO=4:FA=10:GOSUB4500:PRINT"■AGOTADO EL CARGADOR":FORT=0TO4000:NEXT
4060 GOTO420
4100 REM **** PRESENTACION ******
4105 POKEV+42,0:POKEV+43,0:POKE2043,222:POKE2044,223:AC=201
4110 FORCA=40TO108:POKEV+6,CA:POKEV+7,90:AC=AC-1:POKEV+8,AC:POKEV+9,90:NEXT
4115 FORK=204TO680STEP40:FORN=KTOK+18:POKE55296+N,6:NEXT:NEXT
4120 GETC$:IFC$=""THEN4120
4130 FORK=204TO680STEP40:FORN=KTOK+18:POKE1024+N,32:NEXT:NEXT
4200 CO=7:FA=5:GOSUB4500:PRINT"■ABC       ABC"
4210 FA=6:GOSUB4500:PRINT"■DEF       DEF"
4220 FA=7:GOSUB4500:PRINT"■ G         G"
4230 FA=13:GOSUB4500:PRINT"■ABC       ABC"
4240 FA=14:GOSUB4500:PRINT"■DEF       DEF"
4250 FA=15:GOSUB4500:PRINT"■ G         G"
4300 FA=20:CO=1
4500 POKE781,FA:POKE782,CO:POKE783,0:SYS65520:RETURN
4990 RETURN
5000 REM ***** RUTINA CARACTERES *****
5010 FORI=928TO1001:READQ:POKEI,Q:NEXT
5020 PRINTCHR$(142)
5030 POKE52,56:POKE56,56:CLR
5040 FORI=1TO73:READQ:NEXT
5050 POKE56334,PEEK(56334)AND254
5060 POKE1,PEEK(1)AND251
5070 SYS928
5080 POKE1,PEEK(1)OR4
5090 POKE56334,PEEK(56334)OR1
5100 POKE53272,(PEEK(53272)AND240)+14
5110 FORCAR=1TO20
5120 FORBYTE=0TO7:READNUM
5130 POKE14336+(8*CAR)+BYTE,NUM:NEXT:NEXT
5140 REM SPRITES
5150 V=53248:POKEV+21,31:POKE53285,8:POKE53286,9:POKE53276,7
5160 FORR=216TO223
5170 FORI=0TO62:READH:POKER*64+I,H:NEXT:NEXT
```

PROGRAMAS BASIC

```
10000 GOTO15
20000 REM DATAS RUTINA CARACTERES
20010 DATA169,0,141,57,3,169,208,141,56,3,169,0,141,53,3,169,56,141,52,3
20020 DATA173,57,3,133,251,173,56,3,133,252,173,53,3,133,253,173,52,3,133,254
20030 DATA160,0,177,251,145,253,238,57,3,238,53,3,169,0,205,57,3,208,217,238
20040 DATA56,3,238,52,3,169,215,205,56,3,208,204,96
20050 REM DATAS DE CARACTERES
20060 DATA0,0,0,0,0,0,0,8
20070 DATA0,0,0,28,62,62,62,62
20080 DATA0,0,0,0,0,32,112,112
20090 DATA28,28,28,28,31,31,7,0
20100 DATA62,62,63,63,63,254,254,254
20110 DATA112,240,224,192,0,0,0,0
20120 DATA62,62,62,62,62,62,0,0
20130 DATA255,255,255,255,255,255,255,255
20140 DATA0,28,62,62,62,62,0,62
20150 DATA0,0,0,0,3,7,15,15,0,0,0,0,192,224,240,240,15,15,7,3,0,0,0,0
20160 DATA240,240,224,192,0,0,0,0
20170 DATA15,15,15,15,15,15,15,15
20180 DATA255,255,255,255,0,0,0,0
20190 DATA31,31,31,31,31,31,31,31
20200 DATA255,255,255,255,252,248,240,240
20210 DATA240,240,248,252,255,255,255,255
20220 DATA255,255,255,255,63,31,31,31
20230 DATA31,31,31,63,255,255,255,255
20240 REM DATAS VAQUERO IZQ.
20250 DATA0,168,0,0,84,0,2,170,0,0,220,0,0,213,0,0,168,0,0,232,0,0,120,0,3
20260 DATA124,240,3,85,252,0,212,64,0,252,192,0,84,0,0,252,0,0,252,0,0,252,0,0
20270 DATA127,0,0,95,0,1,79,0,1,12,0,2,138,0
20280 REM DATAS VAQUERO DEREC.
20290 DATA0,0,0,0,42,0,0,42,0,0,63,0,0,170,128,0,55,0,0,87,0,0,42,0,0,41,0,0
20300 DATA45,0,42,85,0,10,87,0,2,63,0,0,21,0,0,63,0,0,63,0,0,253,0,0,245
20310 DATA0,0,241,64,0,48,64,0,162,128
20320 REM DATAS DILIGENCIA
20330 DATA0,0,0,3,255,0,13,221,192,15,255,192,15,171,192,14,170,192,14,170,192
20340 DATA15,171,192,15,255,192,5,85,64,199,255,76,199,255,76,207,255,204,207
20350 DATA255,204,247,255,124,247,255,124,205,253,204,207,255,204,192,0,12,192
20360 DATA0,12,0,0,0
20370 REM DATAS VAQUERO IZQ. MUERTO
20380 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,48,0,0
20390 DATA60,0,8,4,48,33,5,252,237,173,252,229,181,204,229,151,204,239,155,5,32
20400 DATA0,0,128,0,0,0,0,0,0,0,0,0,0,0,0
20410 REM DATAS VAQUERO DER. MUERTO
20420 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20430 DATA240,0,0,48,32,3,16,72,79,106,122,127,90,90,127,214,90,115,246,250,0
20440 DATA0,8,0,0,2,0,0,0,0,0,0,0,0,0,0
20450 REM DATAS DISPARO
20460 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20470 DATA0,60,0,0,60,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20480 DATA0,0,0
20490 REM DATAS PRESENTACION
20500 DATA0,0,0,0,0,0,0,0,0,7,224,0,8,16,0,24,24,0,36,36,0,34,68,0
20510 DATA65,130,0,127,254,0,65,130,124,34,68,68,36,36,124,24,24,64,8,16,64
20520 DATA7,224,124,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20530 REM DATAS PRESENTACION
20540 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,1,128,0
20550 DATA249,15,128,129,8,128,249,15,128,9,8,0,9,8,0,249,15,128,0,0,0,0,0,0
20560 DATA0,0,0,0,0,0

READY.
```





STAFF OF KARNTH

La casa de software ULTIMATE PLAY GAME ha creado en juego muy entretenido para nuestro Commodore-64, es el STAFF OF KARNTH. En él encontramos a nuestro protagonista, el noble británico Sir Arthur Pendragón (prestigioso arqueólogo), que descubrió la existencia de un bastón mágico que usaba KARNTH. Este era un malvado brujo que hace cientos de años encontró un globo creado por una diabólica raza que vivió millones de años antes que nosotros los hombres existieramos. Esta diabólica raza de criaturas de recreaba con la tortura y destrucción de otras criaturas, e intentaba dominar el universo mediante una bola de fuego, hecha de un material desconocido en nuestro planeta que provenía del mismo diablo. Esta bola de fuego tenía el poder de transportar criaturas de una dimensión a otra. La maldad de estas criaturas hizo que se enviasen unas a otras a la dimensión alternativa de lo irreal. KARNTH aprendió con los años a usar la terrible bola de fuego y la fusionó con su bastón mágico. El resultado de esta fusión fue encerrado en un obelisco para que fuese acumulando la energía suficiente para poder traer de nuevo a toda la diabólica raza a nuestra dimensión. Al principio, no disponiendo de la energía necesaria, solo pudo traer algunas criaturas para que guardasen el castillo y la llave que abría el obelisco. Pasando el tiempo viendo el brujo que iba a morir y para evitar que alguien estropeara sus planes, rompió la llave en 16 trozos y los distribuyó por el castillo.

A las 12 de la noche, llamada de Walpurgis, el bastón habrá acumulado la suficiente energía para hacer realidad los planes de KARNTH, es decir hacer regresar a la raza de malvados.

Sir Arthur llega al castillo solamente 6 horas antes de que se produzca el fatal acontecimiento. Para evitarlo deberá encontrar los 16 trozos de la llave e insertarlos uno a uno en el obelisco, para que este se abra y así pueda destruirle.

Sir Arthur dispone como ayuda de un anillo del bien, que por medio de unos embrujos le permitirá deshacerse momentaneamente de los seres irreales del castillo.

Como veis es un juego muy entretenido que os hará pasar bastante tiempo delante de la pantalla hasta poder abrir el obelisco.

Los gráficos son bastante buenos y el efecto tridimensional está muy bien conseguido. Espero que os guste también a vosotros.

PISTSTOP II

Es un juego de carreras, en mi opinión el mejor existente para Commodore por el momento. Utiliza el método de ventana, que consiste en que el jugador uno vé por la ventana superior su coche y los obstáculos que le surgen y por la inferior a su oponente. También dispone, cada uno de ellos, de información sobre:

— Velocidad que llevan.
— Tiempo de carrera.
— Combustible que les queda.
— Lugar de la pista donde se encuentra.
— Etc.

La conducción es francamente buena, pues aparte de poder controlar el coche perfectamente con el joy, se tiene posibilidad de meter el «turbo» apretando el disparo.

El juego permite varias opciones, que son:

— Elegir uno o dos jugadores.
— Elegir entre seis circuitos distintos o Grand Circuito, en el cual se puede hacer un campeonato del mundo, al final se realizará una media con los puntos conseguidos en cada uno de los circuitos.
— Elegir entre tres, seis y nueve vueltas por circuito.
— Elegir el tipo de conducción: novato, semiprofesional o profesional.

Otra facilidad que da este juego es la posibilidad de cambiar los neumáticos cuando estos se dañen por los choques contra otros vehículos o roces con el borde de la pista. También se puede llenar el depósito cuando se agote la gasolina.

Este juego os lo recomiendo para vuestra juegoteca.





CLUB DE USUARIOS MONSER

Si ya tienes tu ordenador (SPECTRUM, MSX, C-64) y deseas estar periódicamente informado de los nuevos programas y periféricos, así como recibir todas las ofertas de software y hardware que, con frecuencia hacemos especialmente para nuestros socios, inscríbete hoy mismo sin ningún gasto, ni compromiso por tu parte.

Enviar el cupón a:

CLUB DE USUARIOS MONSER
Calle Argos, 9. 28037-MADRID

Podrás obtener más información a través de las revistas «48K», «Type and Run», «MSX Soft Magazine» y «Commodore Soft Magazine», o llamando al teléfono (91) 742 72 12/96.

Deseo inscribirme en el CLUB DE USUARIOS MONSER sin ningún gasto, ni compromiso, para recibir información y tener acceso a los descuentos y ofertas especiales del Club.

Nombre ....

Dirección.. Teléfono ....

Ciudad.......................... C.P. .... Provincia ....

Edad.................... Profesión ....

Marca del ordenador............................ Fecha compra ....

FIRMA

FECHA

Envía inmediatamente el cupón y recibirás en breve un keypanel de regalo para personalizar tus programas.

La revista Commodore con cassette al mejor precio.

CUPON DE SUSCRIPCION

ahora sólo

5.396 *ptas.*

Envíe HOY MISMO este cupón.
Inmediatamente empezará a recibir sus ejemplares y así durante 1 año (12 ejemplares).

El importe lo abonaré: POR CHEQUE ☐ CONTRA REEMBOLSO ☐ GIRO POSTAL ☐

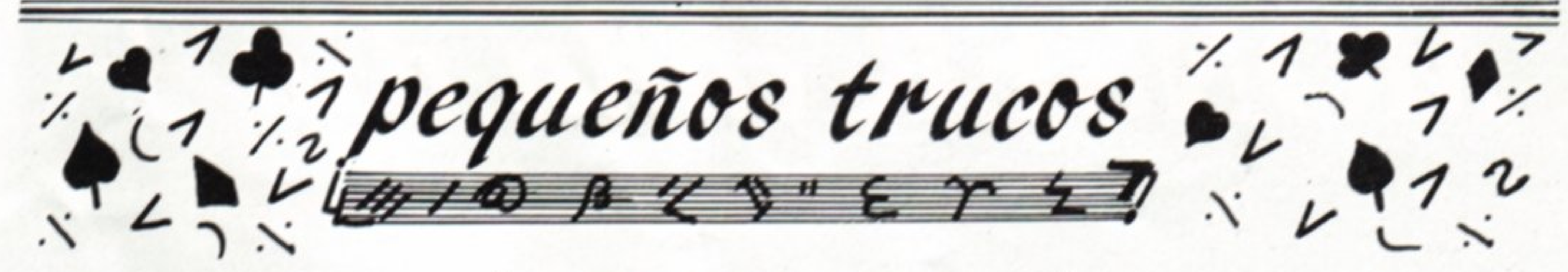
NOMBRE ______________________________

DIRECCION ______________________________

CIUDAD ______________________ C.P. ________

PROVINCIA ______________________________





VELOCIDAD DE LIST

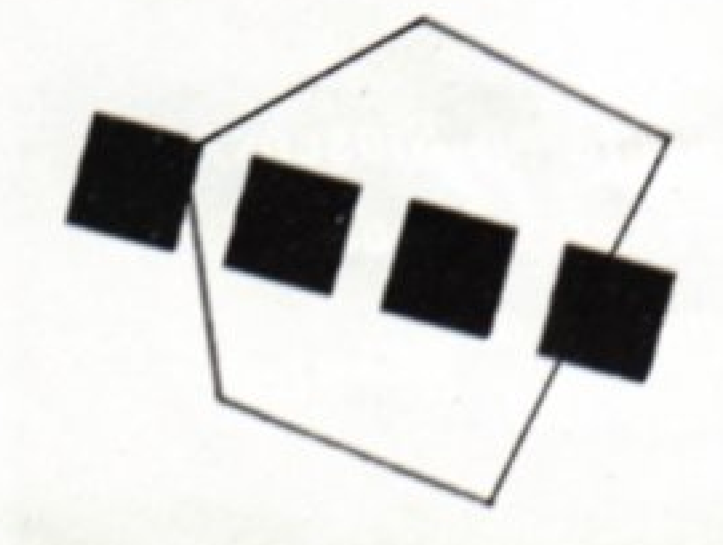
Al efectuar un «LIST» en e Commodore observamos que las líneas pasan a demasiada velocidad ante nuestros ojos, aunque tengamos apretada la tecla «CTRL»; para solucionar este problema deberéis introducir esta pequeña rutina:

```
1   PRINT "VELOCIDAD
    DE LIST (200 a 255)";
2   INPUT VE: POKE 251,
    VE
3   FOR LU: 49152 TO
    49152 + 22: READ
  CO: POKE LU, CO:
NEXT
4      POKE774,0:
POKE775, 192
  100 DATA072, 165, 197,
    201, 004, 240, 250, 201
  200 DATA005, 208, 008,
    165, 251, 133, 162, 165
300 DATA162, 048, 252, 104,
    076, 026, 167
```

Explicación

- Las líneas 1 y 2 ponen la velocidad de scrolling (os recomiendo 255).
- La línea 3 ubica la rutina contenida en las instrucciones DATA a partir de la posición 49152 de memoria, en la cual el BASIC no interfiere.
- La línea 4 altera el vector de la instrucción LIST para que apunte a nuestra rutina.

De esta forma la tecla F1 hará una detención del scrolling y la F3 lo ralentizará.



SALVAR PROGRAMAS EN CODIGO MAQUINA DESDE BASIC

Si alguna vez habéis intentado salvar un programa en código máquina desde BASIC con la instrucción «SAVE "Nombre programa", 1, 1», habréis podido comprobar que además de salvar el programa en código máquina también habréis salvado el programa BASIC que existiera en memoria; esto es debido a que la instrucción «SAVE "Nombre programa", 1, 1» salva desde la posición de inicio de BASIC, la $0801, hasta el final del programa contenido en memoria. Para salvar únicamente el programa en código máquina evitando las posiciones de inicio del BASIC deberéis hacer:

```
PASO 1  LOAD  "Nombre
        programa", 1, 1.
PASO 2  NEW
PASO 3  POKE 43, PEEK
        (829)
PASO 4  POKE 44, PEEK
        (830)
PASO 5  POKE 45, PEEK
        (831)
PASO 6  POKE 46, PEEK
        (832)
PASO 7  SAVE  "Nombre
        de programa"
```

Estos pasos son para programas cargados sin turbo.

Si lo queréis hacer con programas grabados con el turbo deberéis teclear:

```
PASO 1      ← L  «Nombre
programas», 1,1
PASO 2      NEW
PASO 3      POKER 43, PEEK
(828)
PASO 4      POKE 44, PEEK
(829)
PASO 5      POKE 45, PEEK
(830)
PASO 6      POKE 46, PEEK
(831)
PASO 7 ← S "Nombre progra-
ma"
```

BLOQUEO/DESBLOQUEO DE ESCRITURA EN DISCO

Muchos habréis tenido problemas con los papeles adhesivos que tapan la ventana del disco, para impedir su escritura. para evitar tener que poner este adhesivo podeis emplear el programa que a continuación se detalla, el cual cambia el segundo byte del RAM del disco por un «1» (con valor ASSCII) que hace que el sistema operativo pueda leer el disco pero no escribir en él. Por el contrario si queremos escribir pondrá una «A» (valor ASSCII).

```
 1  OPEN   15,8,15,  "I0":
OPEN 2,8,2,"#"
 2  PRINT#
15,"U1: 2;0;18;0:GOSUB20
 3  PRINT#15,"B-P:";2;2
 4  INPUT«(B)LOQUEO/
(D)ESBLOQUEO»;A$:IF
A$ = «D»THEN 7
 5  IF    A$ < >«B»THEN»
ERROR EN INPUT:GOTO 4
 6  PRINT#2,CHR$ (1);:GO-
SUB20:GOTO10
 7  PRINT # 2, CHR$ (65);
GOSUB20
 8  PRINT # 15, «M-W»;
CHR$ (1); CHR$ (1); CHR$ (1);
CHR$ (65)
 9  PRINT # 15, «M-W»;
CHR$ (2); CHR$ (7); CHR$ (1);
CHR$ (65)
10  PRINT # 15, «V2»;
2;0;18;0: GOSUB20
11  CLOSE2: PRINT # 15,
«I0», CLOSE15: END
20  INPUT#15, EN, EM$, ET,
ES
21  IFENTHENPRINT
«ERROR DISCO» EN; EM$;
ET; ES
22  RETURN
```

La única desventaja de este sistema es que el disco admite el formateo.

MONSER cada dia +



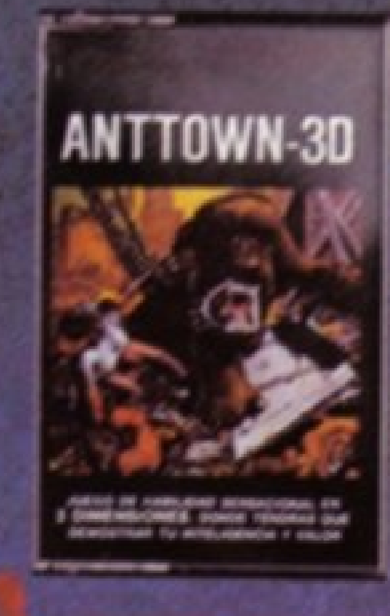
6 SuperCasettes
Full Memory
por solo 1.795 pts.

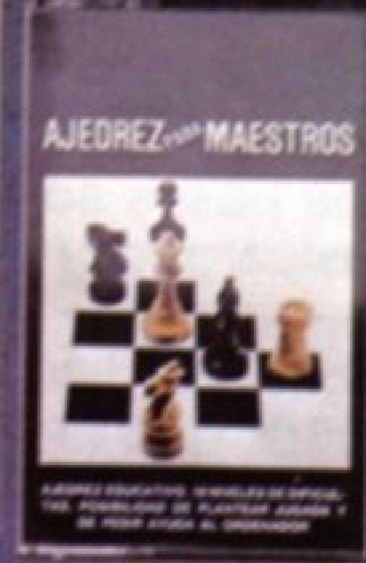


Libreto que contiene manual
de instrucciones en Castellano
+Aplicaciones para tu Spectrum















MONSER S. A.
C/ Argos, 9 - 28037 Madrid .
Teléfonos: (91) 742 72 12 - 742 72 96

Ya a la venta en Kioscos,
Tiendas Especializadas y
Departamento de
Informatica del El Corte Inglés

curso código máquina



Continuamos nuestro curso de código máquina haciendo una breve descripción de cada uno de los registros de la CPU.

— ACUMULADOR (A). Es el registro de trabajo principal. Como sabemos, sirve para almacenar el operando y es donde se almacena el resultado de las operaciones.

— REGISTROS DE INDICE X e Y. Son registros auxiliares de 8 bits. Su característica especial de poder incrementarse o decrementarse mediante instrucciones específicas les hace muy útiles para el manejo de tablas.

— CONTADOR DE PROGRAMA (PC). Está compuesto por dos registros de 8 bits, PCL y PCH. Contiene, como su nombre indica, la dirección de memoria donde está la próxima instrucción a ejecutarse. Una propiedad esencial de éste registro es la posibilidad de autoincrementarse bajo una orden procedente del controlador interno.

— PUNTERO DEL STACK (SP). Es un registro de 8 bits que permite controlar una zona de memoria especial denominada Stack y de una página de tamaño (256 posiciones de memoria). El Stack es una pila LIFO, es decir el último en entrar es el primero en salir. Para explicar que es el Stack imaginemos un montón de libros uno encima de otro, si ponemos un nuevo libro éste queda encima de todos ellos y por tanto al ir a quitar uno será el primero que cojamos, y si queremos uno intermedio para llegar a él debemos retirar primero todos los anteriores.

Para saber que posición de memoria es la primera de la pila de datos usamos el puntero del stack, donde sus 8 bits representan los 8 bits más bajos de la dirección. Los 8 bits más altos, para ésta CPU, tienen un valor fijo de 1. Por tanto, la estructura de este registro obliga a tener el Stack siempre en la página 1, es decir desde la posición 256 hasta la 511.

— REGISTRO DE STATUS (P). Su estructura se muestra en la figura 4. Como puede verse el bit 5 no se utiliza. Vamos a analizar cada uno de los bits de este registro.

— BIT DE ACARREO (C). Indica si la operación efectuada tiene o no acarreo, es decir si el resultado es superior o no a 255.

BIT DE CERO (Z). Indica que el resultado de una operación es cero, y lo hace poniéndose a 1.

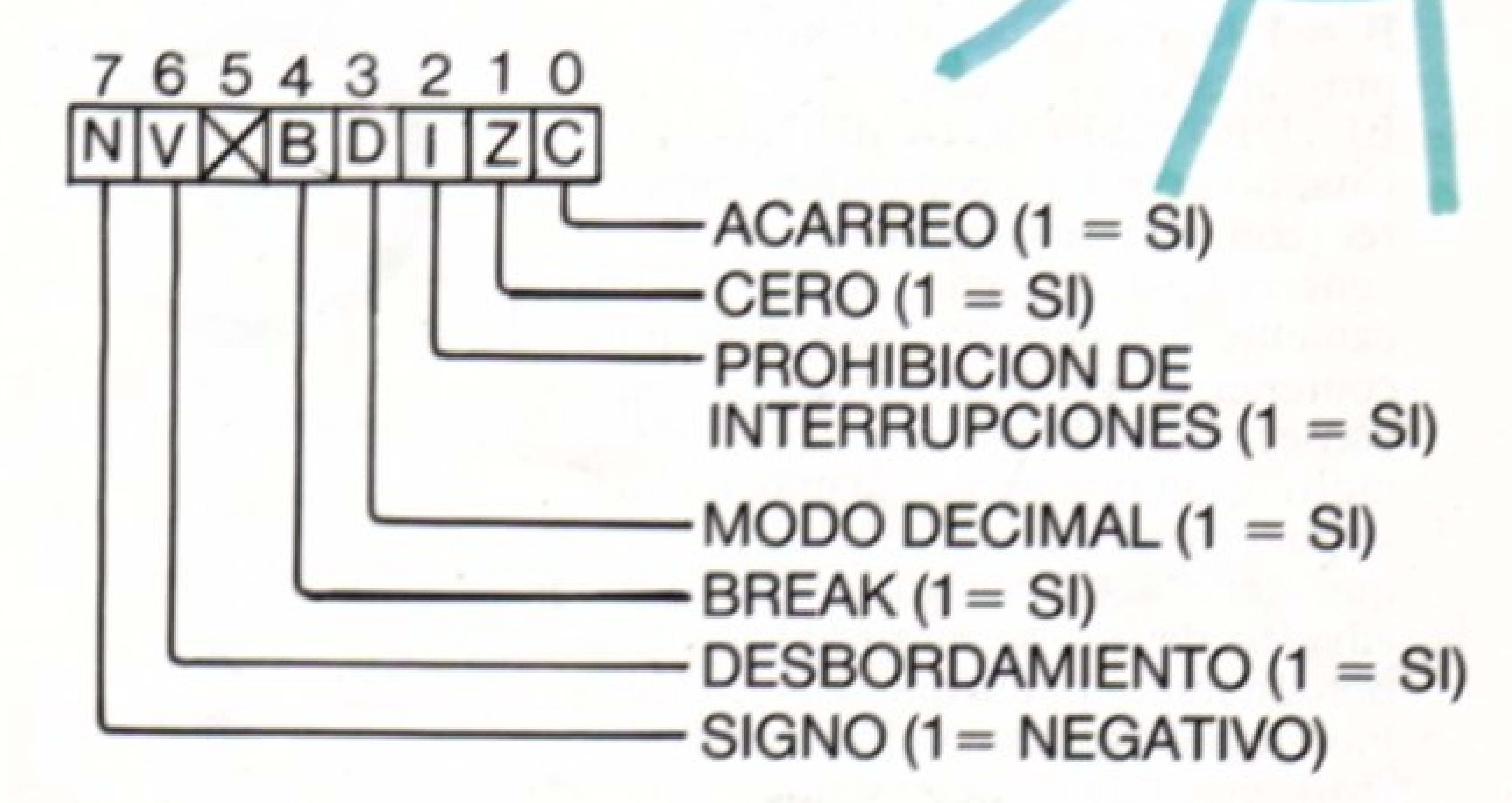


Figura 4. Estructura del registro de Status

— BIT DE PROHIBICION DE INTERRUPCIONES (I). Afecta a las peticiones que llegan a través del pin IRQ del chip 6510 que serán atendidas solo si I = 0.

— BIT DE MODO DECIMAL (D). Permite efecutar sumas y restas en los que los operandos son dos números decimales, ésto ocurre cuando D = 1.
Vamos a ver un ejemplo para comprobar el funcionamiento.

	Modo normal		Modo decimal	
		(Hex)		(Dec)
Primer sumando Segundo sumando	00101000 00110110	28 36	00101000 00110110	28 36
Suma	01011110	5E	01100100	64

En el modo normal se interpreta que los sumandos son 28 + 36 (hexadecimal) y el resultado es 5E. En el modo decimal se interpreta que los sumandos son 28 + 36 (decimal) y el resultado es 64 (decimal).

— BIT DE BREAK (B). Este bit sirve para identificar si la interrupción ha sido realizada mediante una entrada en el pin de IRQ o mediante programa. Si B = 1 la interrupción la provocó el programa en ejecución.

— BIT DE DESBORDAMIENTO (V). Cuando se trabaja con códigos bipolares (con números negativos), el bit 7 se conserva para el signo, quedando únicamente los siete bits restantes para contener el valor absoluto. Si como consecuencia de una operación, el número resultante excede al máximo que se puede representar con 7 bits, con lo que afecta al bit de signo, es necesario advertir de que se ha producido desbordamiento, por lo que se pone V a 1. Para la lógica sin signo (números positivos) este flag no tiene sentido.

— BIT DE SIGNO (N). Indica si el resultado de la operación es positivo o negativo.

SISTEMA HEXADECIMAL DE NUMERACION

En nuestra vida cotidiana utilizamos el sistema decimal de numeración para representar los números. Como su nombre indica, existen diez dígitos base (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) con los que podemos representar cualquier número por grande que éste sea.

En el mundo de los ordenadores este sistema no es posible utilizarlo, por lo menos sería muy difícil implantarlo. La electrónica digital, como ya sabemos, se basa en los dos estados posibles de un bit, el 0 y el 1. Con estos dos dígitos base se formó el sistema binario de numeración que representa cualquier número mediante la combinación de ceros y unos.

El sistema binario, como se puede imaginar, es muy engorroso de utilizar en cuanto los números tratados son un poco grandes por la cantidad de ceros y unos que aparecen. Como muestra de ello tenemos los siguientes ejemplos:

$$20_{10} = 10100_2$$
$$80_{10} = 1010110_2$$
$$654_{10} = 1010001110_2$$

Para facilitar el manejo de los números binarios se desarrolló el sistema hexadecimal de numeración, que consiste en agrupar de cuatro en cuatro los dígitos binarios y representarlos por un único carácter hexadecimal. Al agrupar cuatro dígitos que pueden tomar dos valores tenemos 16 combinaciones diferentes, es decir, el sistema hexadecimal consta de 16 elementos. Como sólo existen diez números el resto se representan con letras de la A a la F. La tabla 1 define los dígitos hexadecimales y sus equivalentes binarios y decimales.

A partir de ahora utilizaremos el símbolo «$» para representar que el número está en notación hexadecimal, el símbolo «%» para la notación binaria y ningún símbolo para la notación decimal.

Dígito hexadecimal	Binario equivalente	Decimal equivalente
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Tabla 1. Tabla de dígitos hexadecimales.

En los microprocesadores de 8 bits el bus de datos consta de 8 bits, por lo que los datos se pueden representar con dos dígitos hexadecimales. Los datos pueden valer desde $00 hasta $FF, es decir, de 0 a 255. Como sabemos, el bus de direcciones consta de 16 bits, por lo que necesitaremos 4 dígitos para representarlas. Las direcciones van desde $0000 hasta la $FFFF, es decir, desde la 0 hasta la 65535.

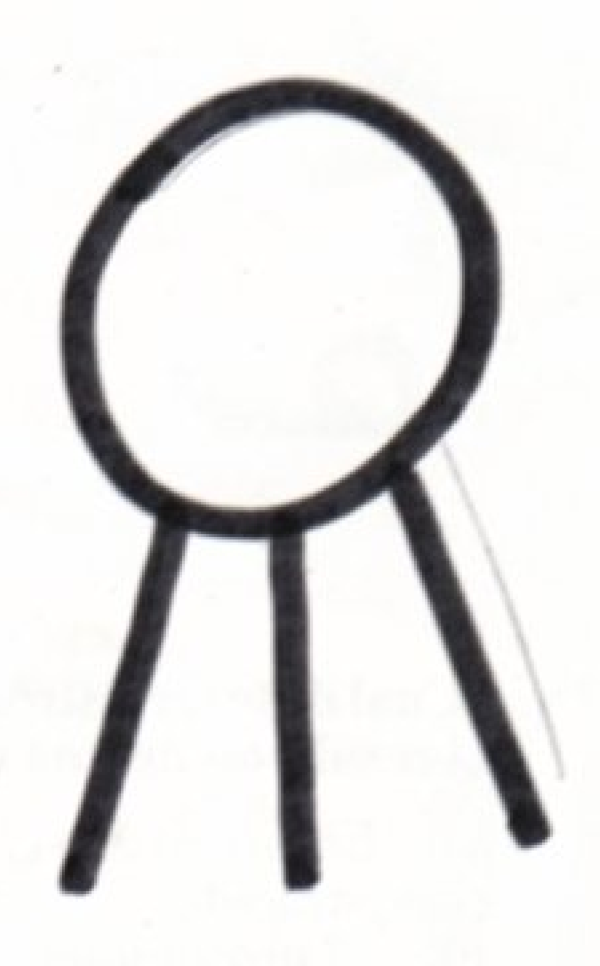
Vamos a ver ahora cómo se hace la conversión de uno a otro sistema. Si tenemos el número hexadecimal $ABCD, para pasarlo a decimal deberemos utilizar la fórmula:

$$N_{10} = A \times 16^3 + B \times 16^2 + C \times 16^1 + D$$

Pero para mayor facilidad en el paso de uno a otro sistema tenemos la tabla 2.

Vamos a explicar su funcionamiento mediante unos ejemplos. Supongamos que tenemos el número hexadecimal $85EA y queremos calcular el equivalente decimal, para ello nos vamos a la tabla 2 y en la columna 4 para el $8 tenemos el valor 32768, en la columna 3 para el $5 tenemos el valor 1280, en la 2 para el $E tenemos 224 y en la 1 para el $A tenemos 10. Con esto el valor decimal buscado es:

$$34282 = 32768 + 1280 + 224 + 10$$



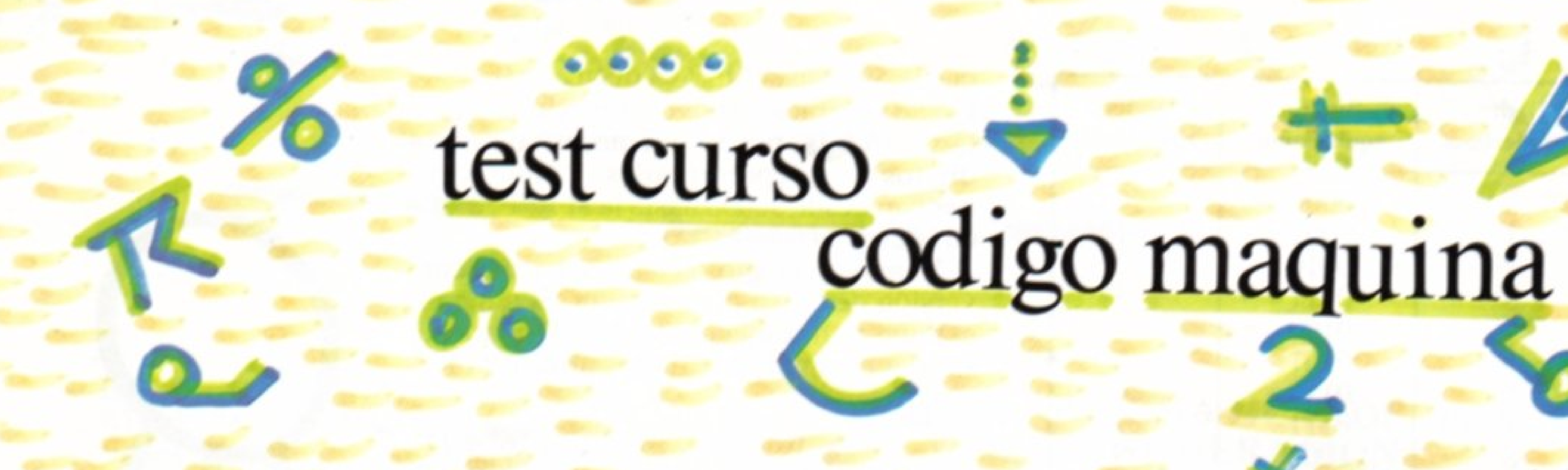
COLUMNAS HEXADECIMALES									
5		4		3		2		1	
Hex.	Dec.	Hex.	Dec.	Hex.	Dec.	Hex.	Dec.	Hex.	Dec.
0	0	0	0	0	0	0	0	0	0
1	65536	1	4096	1	256	1	16	1	1
2	131072	2	8192	2	512	2	32	2	2
3	196608	3	12288	3	768	3	48	3	3
4	262144	4	16384	4	1024	4	64	4	4
5	327680	5	20480	5	1280	5	80	5	5
6	393216	6	24576	6	1536	6	96	6	6
7	458752	7	28672	7	1792	7	112	7	7
8	524288	8	32768	8	2048	8	128	8	8
9	589824	9	36864	9	2304	9	144	9	9
A	655360	A	40960	A	2560	A	160	A	10
B	720896	B	45056	B	2816	B	176	B	11
C	786432	C	49152	C	3072	C	192	C	12
D	851968	D	53248	D	3328	D	208	D	13
E	917504	E	57344	E	3584	E	224	E	14
F	983040	F	61440	F	3840	F	240	F	15



Supongamos ahora que tenemos el número decimal 39653 y queremos calcular su equivalente hexadecimal. Para hacerlo seguimos el camino inverso al anterior. Tomamos la tabla 2 y en la columna 4 el número inmediatamente inferior al dado es 36864 que corresponde al $9. Con esto nos queda 2789 = 39653 − 36864. En la columna 3 el número inmediatamente inferior a 2789 es 2560, que corresponde a $A. Con esto queda 229 = 2789 − 2560. En la columna 2 el número inmediatamente inferior a 229 es el 224, que corresponde a $E, y sobra 5 que corresponde a $5 en la columna 1. Entonces 36864 = $9AE5.

Para hacer operaciones aritméticas con números hexadecimales os recomiendo que los paséis a decimal, hagáis la operación y entonces pasar el resultado a hexadecimal; aunque parezca más largo, si no tenéis soltura, seguro que tardáis más haciendo la operación directamente.





1) ¿Cúal de los registros de la CPU indica el resultado de una comparación?

a) El egistro X o el Y según el tipo de comparación.
b) El acumulador.
c) El puntero de stack.
d) El registro de estado.
e) Ninguna de las anteriores.

2) ¿En qué página se encuentra el stack de la CPU?

a) En la página 16
b) En la página 0
c) Entre la página 0 y la 1
d) En la página 1
e) No está en ninguna página.

3) El bit D cuando esta a 0 permite:

a) Sumas y restas en decimal.
b) No permite sumas ni restas.
c) Sumas y restas en hexadecimal.
d) a) y b)
e) Ninguna de las anteriores.

4) La notación binaria se suele indicar:

a) Con un $ delante del número.
b) Con un @ delante del número.
c) Con un % delante del número.
d) No hace falta indicarlo.
e) Ninguna de las anteriores.

5) El contador de programas cuando tiene el valor hexadecimal $ 1000 indica:

a) Que la instrucción que se está ejecutando esta en la posición $ 1000 de memoria.
b) Que la instrucción que se va a ejecutar está en la posición $ 1000 de memoria.
c) Que el contenido de la posición $ 1000 nos indica la dirección del programa.
d) No indica nada porque siempre toma ese valor por defecto.
e) Ninguna de las anteriores.

6) El PCH:

a) Es un registro de la CPU.
b) No se encuentra en la CPU.
c) Es parte de un registro de la CPU.
d) Es parte del contador de programa.
e) c) y d).

7) El bit de desbordamiento:

a) Sirve para indicar que se trabaja con números negativos.
b) Es el bit V.
c) Avisa de que el número excede de los 7 bits.
d) Se encuentra en el registro de STATUS.
e) Todas las anteriores.

8) Que es una RON (Repaso mirar número anterior):

a) Es una memoria de sólo lectura.
b) Es una memoria de sólo escritura.
c) No es una memoria.
d) Es una memoria de lectura/escritura.
e) Ninguna de las anteriores.

9) La representación del número $ 36 es:

a) Su decimal 1024.
b) En binario 01001010.
c) En decimal 40.
d) En binario 01001011.
e) Ninguna de las anteriores.

10) El stack se encuentra:

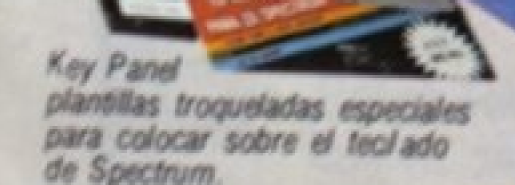
a) En la primera página de memoria.
b) Entra las posiciones 255 a 510 de memoria.
c) Entre las posiciones 256 a 511 de memoria.
d) Entre las posiciones $ 0100 a $ 01FF.
e) a) c) y d).



SOLUCIONES

1.ª d)
2.ª d)
3.ª c)
4.ª c)
5.ª b)
6.ª e)
7.ª e)
8.ª a)
9.ª b)
10.ª e)

MONSER... cada dia mas.



Key Panel
plantillas troqueladas especiales para colocar sobre el teclado de Spectrum.

Games Board
carcasa moldeada para colocar sobre el teclado del Spectrum, con pivotes móviles para indicar las teclas a utilizar en cada programa.

Revista con cassette de juegos de aparición mensual con programas didácticos para Spectrum 48 K

Colección tus Juegos Single
20 estuches de lujo con un programa para Spectrum 48 K incluyendo manual en castellano.

Colección tus juegos 4
5 estuches de lujo con 4 programas cada uno incluyendo manual de instrucciones en castellano para Spectrum 48 K.



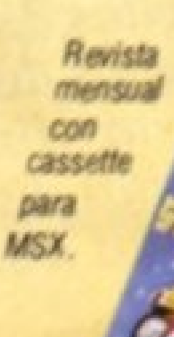
Revista con cassette de juegos de aparición mensual para Spectrum 48 K.

Seis cassettes con programas estrellas presentados en estuche de lujo para Spectrum 48 K
P.V.P. 1.795 pts.
Catálogo con instrucciones en castellano.

MONSER

MONSER, S.A.
C/ Argos, 9
28037 Madrid
Tlf. 742 72 12 / 96





Revista mensual con programas listados para Spectrum, M S X, Amstrad y Commodore, incluyendo cinta virgen.

Revista mensual con cassette para MSX.



Interface para Joystick de doble salida, especial para Spectrum.

Lápiz de Luz para Spectrum.



Joystick Quinckshot I



Cassette virgen C-30, especial para ordenadores.

Dos programas de juegos para Spectrum 48 K, con cinta virgen de regalo.



Joystick Quickshot II



Joystick «Challenger»

Key Board
teclado profesional especial para ordenadores.

Nota: Todos los cassettes llevan los programas grabados en ambas caras.

CONCURSO BASIC

1.ª) ¿Cuantos elementos tendrá un array que tiene 8 filas y 5 columnas?

a) 13 elementos.
b) 38 elementos.
c) 40 elementos.
d) 41 elementos.
e) Ninguna de las anteriores.

2.ª) ¿Que significado tiene A % (1,2) = 5?

a) El valor asignado al elemento de la primera fila y segunda columna es 5.
b) El valor asignado al elemento de la segunda fila y primer columna es 5.
c) Que a todos los elementos del array se les asigna el valor 5.
d) Que es un string que vale 5.
e) Ninguna de las anteriores.

3.ª) Un array hace falta declararlo mediante un DIM:

a) Siempre.
b) Sólo si se trata de un array de cadena.
c) Sólo si tiene más de una dimensión y el número de elementos excede de 11
d) No es necesario aunque exceda de 11 elementos.
e) Ninguna de las anteriores.

4.ª) Si declaramos una tabla con 15 elementos y escribimos A (20) = 2.5.

a) El intérprete nos dará un mensaje de error.
b) El intérprete se saltará esa instrucción.
c) El intérprete ampliará el número máximo de elementos a 21.
d) El sistema se quedará colgado.
e) Ninguna de las anteriores.

5.ª) ¿Cúal de las siguientes afirmaciones es correcta?

a) Un array y una variable no tiene nada que ver
b) Un arry y una variable son lo mismo
c) Un array unidimensional de un elemento es una variable.
d) Un array bidimensional solo puede contener un variable.
e) Ninguna de las anteriores.

CONCURSO CODIGO MAQUINA

1.º) El bit B (Break) sirve para

a) Avisar que la CPU ha sido interrumpida.
b) Si ha habido una interrupción, saber si es por programa o no.
c) La a) y b).
d) Saber si va a haber interrupción de CPU.
e) Ninguna de las anteriores.

2) ¿El número $ 1000 tiene como valor equivalente:

a) 4095 en decimal.
b) 4096 en decimal.
c) 0001000000000000 en binario.
d) 000100000000 en binario.
e) La b) y c).

3) El bit I sirve para:

a) Atender las interrupciones si está a 1.
b) Atender las interrupciones si está a 0.
c) No atender las interrupciones si está a 0.
d) No atender las interrupciones si está a 1.
e) La b) y c).

4) El sistema de numeración en base 16 contiene:

a) 16 dígitos distintos.
b) 15 dígitos distintos.
c) 14 dígitos distintos.
d) 13 dígitos distintos.
e) 12 dígitos distintos.

5) El registro de status tiene:

a) 5 bits.
b) 6 bits.
c) 7 bits.
d) 8 bits.
e) ninguna.

Concurso de Programas listados

El presente concurso está abierto a todos nuestros lectores. La inscripción y participación es gratuita.

El concurso será mensual.

El importe del premio según valoración de nuestro jurado, nombrado al efecto por MONSER, oscilará entre 5.000 y 15.000 pesetas.

Entre todos los programas recibidos, aunque no hayan sido premiados, se realizará a finales de año un sorteo, obteniendo el afortunado ganador un gran premio, a elegir entre una Unidad de Disco, una Impresora o un Monitor Color. Por este motivo las cintas en ningún caso serán devueltas.

Los nombres de los concursantes premiados se publicarán en la revista.

BASES

No se establece ninguna limitación en cuanto a temática y extensión del programa, tan sólo que deben estar destinados al COMMODORE-64.

Los participantes deberán enviarnos un cassette o diskette conteniendo el programa, una explicación del mismo y si es posible un listado.

Los programas seleccionados serán publicados en la revista, quedando todos los derechos de éstos en propiedad de MONSER.

GANADOR DEL CONCURSO COMMODORE

EL GANADOR DE NUESTRO CONCURSO SOBRE PREGUNTAS Y BASIC HA SIDO:
JOSE M.ª MONELL GUIX
C/ RAMON FOLCH II, 12
CARDONA (BARCELONA)

ROGAMOS A NUESTRO GANADOR SE PONGA EN CONTACTO CON NOSOTROS, PARA QUE NUESTRO DEPARTAMENTO DE ADMINISTRACION LE HAGA SUBCRITOR DE NUESTRA REVISTA.



LAS RESPUESTAS CORRECTAS DEL N.º 1 SON:

CODIGO MAQUINA	BASIC
1 — E	1 — C
2 — C	2 — B ó C
3 — B	3 — C
4 — C	4 — A
5 — C	5 — C ó E

CONCURSO DE CODIGO MAQUINA Y BASIC

La inscripción en el concurso será gratuita.

Se realizará un sorteo «todos los meses» entre los acertantes a nuestras preguntas.

El premio consistirá en una suscripción a la revista y si el participante ya es suscriptor, será obsequiado con una gran set de juegos.

El nombre del concursante premiado cada mes aparecerá en la revista junto con las respuestas acertadas.

CONCURSO DE CODIGO MAQUINA Y BASIC

NOMBRE ____________________

DIRECCION ____________________

CIUDAD ____________________ C.P. ________

PROVINCIA ____________________ EDAD ________

Preguntas Codigo Maquina				
1	2	3	4	5

Preguntas Basic				
1	2	3	4	5

Ponga la respuesta adecuada en cada casilla y enviar a:

MONSER C/ Argos. 9 28037 Madrid